Research Report

Report 6

MTRAC—A COMPUTER PROGRAM FOR ANALYSIS OF CIRCUITS INCLUDING MAGNETIC CORES

Volume I

Computation, Program, and Application

By: D. NITZAN and J. R. HERNDON

Prepared for:

JET PROPULSION LABORATORY
4800 OAK GROVE DRIVE
PASADENA, CALIFORNIA 91103

CONTRACT 951840 UNDER NAS7-100

(Extension of work under Contracts 950095 under NASw-6 and 950943 and 951383 under NAS7-100)

This work was performed for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration under Contract NAS7-100.

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 • U.S.A.

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

Report 6 *June 1969*

MTRAC—A COMPUTER PROGRAM FOR ANALYSIS OF CIRCUITS INCLUDING MAGNETIC CORES

Volume I

Computation, Program, and Application

By: D. NITZAN and J. R. HERNDON

Prepared for:

JET PROPULSION LABORATORY
4800 OAK GROVE DRIVE
PASADENA, CALIFORNIA 91103

CONTRACT 951840 UNDER NAS7-100

(Extension of work under Contracts 950095 under NASw-6 and 950943 and 951383 under NAS7-100)

SRI Project 6408

Approved:

F. J. KAMPHOEFNER
Director
Engineering Sciences Laboratory

TORBEN MEISLING
Executive Director
Information Science and Engineering

ABSTRACT

Transient analysis of circuits that include magnetic cores is complicated by the nonlinearity of the switching-core model and by the magnetic coupling among the loop currents. These difficulties have been overcome by incorporating static and dynamic core models into the automated circuit-analysis computer-program TRAC. The program has been modified by including provisions for successive modes of operation, conditional monitor printout of variable values in different portions of the program, plot and print-plot routines with external or internal specifications for scales, units, and frame sizes, normal and circuit-failure run termination, nonlinear inductor and resistor models, etc. The modified computer program is named MTRAC. Time variables (voltages, currents, MMFs, fluxes, etc.) of complex magnetic-core circuits (up to 60 nodes) can thus be computed and plotted automatically. All that the user has to supply is the general run-control specifications, the circuit topology, and the values of the circuit-element parameters.

The computation in MTRAC is iterative. Its method is based on expressing the current through any circuit element connected between Nodes a and b as $i_{ab} = \Delta H(V_a - V_b) - \Delta T$, where ΔH and ΔT are constants. Applying this relation to every circuit node yields a matrix equation $T] = [H] \times V]$, where $V]$ is a matrix whose elements are the unknown nodal voltages. The solution of $V]$ is used to compute the currents through each nonlinear circuit element, and the results are tested for convergence. Expressions are derived for ΔH and ΔT of sources (current or voltage), resistors, capacitors, inductors, zener diodes, diodes, transistors, and magnetic cores.

The MTRAC program consists of two sections, one dealing with initialization, and the other with the transient solution. The initialization section performs the following five tasks: (1) Read in and print out general input data. (2) Read in and print out the circuit-element data. (3) Solve initial conditions (optional). (4) Print out and store

initial conditions. (5) Read in and print out continued-run data (optional). The transient-solution section performs the following seven tasks: (1) Compute the magnitudes of the time-variable current and voltage sources. (2) Until convergence is achieved, compute by iterations (using a routine for solving matrix equations and a modified Newton-Raphson method for solving transcendental equations) all the nodal voltages and all the currents through the nonlinear elements (diodes, transistors, and magnetic-core windings); if necessary reset the unknown values and cut the time step, Δt. (3) Compute the currents through the linear elements. (4) Adjust Δt according to the recent convergence conditions and update the time variables for the next Δt. (5) Store (for plots) and print out the resulting time variables. (6) If the run-time limit is about to be exceeded, punch the final results necessary for a future continued run on cards. (7) Print-plot the specified variable waveforms and store the plot data on a tape; begin a new mode, or exit.

Instructions for data entry by the user are provided. These include definitions of special functions and/or auxiliary variables, and input-data cards specifying the run control and the circuit-element topology and parameter values. The MTRAC program has been applied successfully to transient analyses of several magnetic-core circuits. Resulting waveforms are illustrated for three circuits: a code-diode shift register, a two-phase current driver, and an information-sensing driver. Experimental verification is provided for the first two circuits.

CONTENTS

CONTENTS

ILLUSTRATIONS

TABLES

PREFACE

This is the final report in a series of yearly projects investigating the properties of square-loop magnetic cores for the Jet Propulsion Laboratory. This investigation has covered basic physics of magnetism, flux-switching behavior, magnetic core modeling, and the application of these models to computer circuit analysis. As a result of this investigation, five technical reports[1-5*] (referred to as Reports 1 through 5) and nine technical papers[6-14] have been published.

As the complexity of the magnetic circuits to be analyzed has increased steadily, it was decided at the outset of this project to incorporate the core models into an *automated* circuit-transient-analysis computer program. Autonetics' program TRAC (Transient Radiation Analysis by Computer)[15-18] was selected because of its efficiency and its ready availability. In the course of implementing this objective, several modifications have been made in the original TRAC program.

The modified program, called MTRAC, is described in this report. For convenience, the report is divided into two volumes. Volume I describes the method of computation, the organization, and the application of the MTRAC program. Volume II includes instructions for the input-data cards, glossary of variables, and the listing of the MTRAC program.

* References are listed at the end of Volume I.

I METHOD OF COMPUTATION IN MTRAC

The basic computation method used in the MTRAC computer program is described first. It will be shown that the unknown circuit time variables can be solved from the matrix equation $T] = [H] \times V]$, where $T]$ and $V]$ are column matrices of currents and nodal voltages, respectively, and $[H]$ is a square conductance matrix. On the basis of this method, the contributions to the $[H]$ and $T]$ matrices will then be derived for various circuit elements, such as a current source, a resistor, a voltage source with internal series resistance, a capacitor, an inductor, a zener diode, a diode, a transistor, and a square-loop magnetic core.

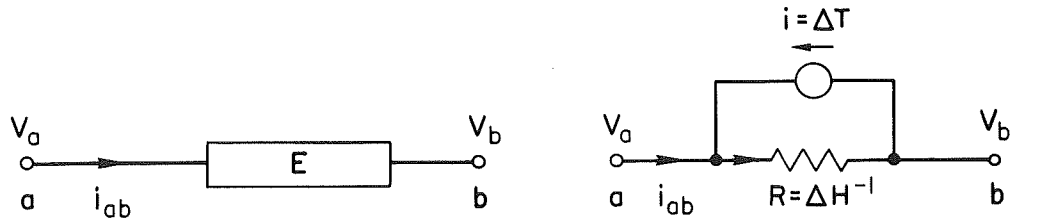
A. Basic Computation Method

1. Matrix Equation $T] = [H] \times V]$

a. Derivation of $T] = [H] \times V]$

Consider a linear or nonlinear two-terminal circuit element E, Fig. 1(a). The element is connected between two nodes, a and b, whose voltages (to ground) are V_a and V_b, respectively. As we shall show later, the current from Node a to Node b at a given time step can be approximated by the linear expression

$$i_{ab} = \Delta H(V_a - V_b) - \Delta T \quad , \qquad (1)$$



(a) SCHEMATIC REPRESENTATION (b) APPROXIMATE EQUIVALENT ELEMENT

TA-6408-1

Figure 1 A TWO-TERMINAL CIRCUIT ELEMENT

where ΔH and ΔT are constants representing conductance and current, respectively. Thus, the approximate equivalence of a two-terminal element is a conductance ΔH in parallel with a current source ΔT, as shown in Fig. 1(b). In the case of a nonlinear circuit element, Eq. (1) is based on "linearization" of the element's operational (or dynamic) characteristics.

Now suppose that n_p two-terminal circuit elements are connected in parallel between Nodes a and b, as shown in Fig. 2. These elements may be represented by an equivalent two-terminal circuit element, enclosed by the dashed line in Fig. 2, whose current is

$$i_{ab} = \sum_{j=1}^{n_p} i_j \quad . \tag{2}$$

Applying Eq. (1) to each jth current in Eq. (2) gives

$$i_{ab} = \sum_{j=1}^{n_p} \Delta H_j (V_a - V_b) - \sum_{j=1}^{n_p} \Delta T_j \quad .$$

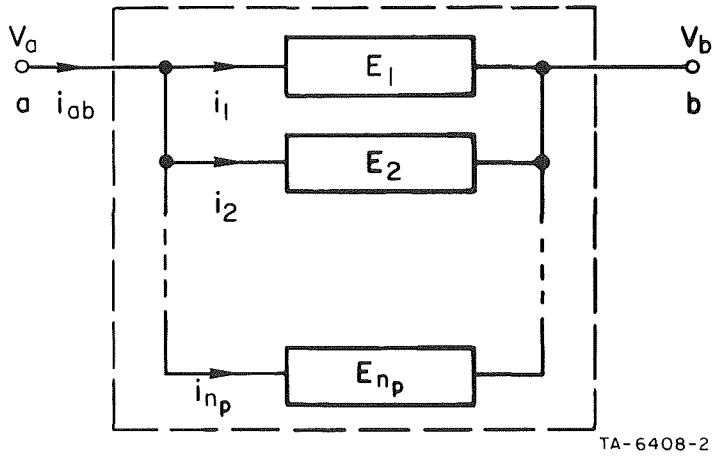


Figure 2 TWO-TERMINAL CIRCUIT ELEMENTS IN PARALLEL

Hence, ΔH and ΔT of the equivalent circuit element are

$$\Delta H_{(p)} = \sum_{j=1}^{n_p} \Delta H_j \tag{3}$$

and

$$\Delta T_{(p)} = \sum_{j=1}^{n_p} \Delta T_j \quad . \tag{4}$$

If the n_p elements are identical, then Eqs. (3) and (4) are reduced to

$$\Delta H_{(p,i)} = n_p \Delta H \tag{5}$$

and

$$\Delta T_{(p,i)} = n_p \Delta T \quad , \tag{6}$$

where the subscript i is added to designate *identical* elements and where ΔH and ΔT correspond to each individual element.

Consider now n_s identical two-terminal circuit elements which are connected in series between Nodes a and b, as shown in Fig. 3. The current through each element is i_{ab}. Applying Eq. (1) to the current of each element, we obtain

$$\begin{aligned} i_{ab} &= \Delta H(V_a - V_1) - \Delta T \\ i_{ab} &= \Delta H(V_1 - V_2) - \Delta T \\ &\vdots \qquad\qquad \vdots \\ i_{ab} &= \Delta H(V_{n_s-1} - V_b) - \Delta T \quad . \end{aligned}$$

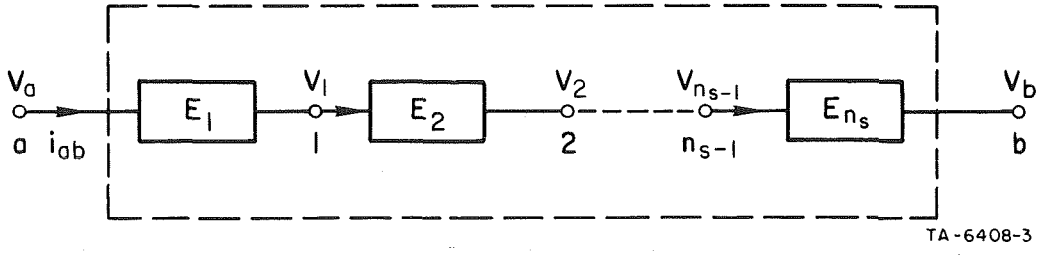


Figure 3 IDENTICAL TWO-TERMINAL CIRCUIT ELEMENTS IN SERIES

Adding these n_s equations and dividing by n_s gives

$$i_{ab} = \frac{\Delta H}{n_s}(V_a - V_b) - \Delta T \quad . \tag{7}$$

Identifying Eq. (7) with Eq. (1) gives ΔH and ΔT of an equivalent element of n_s identical two-terminal elements in series:

$$\Delta H_{(s,i)} = \Delta H/n_s \tag{8}$$

and

$$\Delta T_{(s,i)} = \Delta T \quad . \tag{9}$$

Comparing the equivalent circuits of elements connected in parallel and in series, we observe the following:

(1) Unlike the case of elements in parallel, the current i_{ab} through an equivalent element of n_s *different* elements in series cannot be brought into the form of Eq. (1). Consequently, we can represent n_s elements in series by an equivalent element only if they are identical.

(2) Physically, Eqs. (5) and (8) simply state that the conductance of n_p identical elements in parallel is n_p times larger than and the conductance of n_s identical elements in series is n_s times smaller than the conductance of an individual element.

(3) Equations (6) and (9) agree with the fact that currents add in a parallel connection, but are the same in a series connection.

Consider now an entire network of n nodes, any one of which may be Node a, *i.e.*, $1 \leq a \leq n$. Any jth element between Node a and the remaining $n - 1$ nodes is represented by Node b in Eq. (1). For each of these $n - 1$ elements, Eq. (1) is written as

$$i_{aj} = \Delta H_j(V_a - V_j) - \Delta T_j \quad . \tag{10}$$

If there is no element between Node a and, say, Node m, then $\Delta H_m = 0$ and $\Delta T_m = 0$; hence, $i_{am} = 0$, as expected. For reference, let us assume that all the $n - 1$ currents *leave* Node a. An example for the case of $a = 1$ is shown schematically in Fig. 4. Applying Eq. (10) to each of the

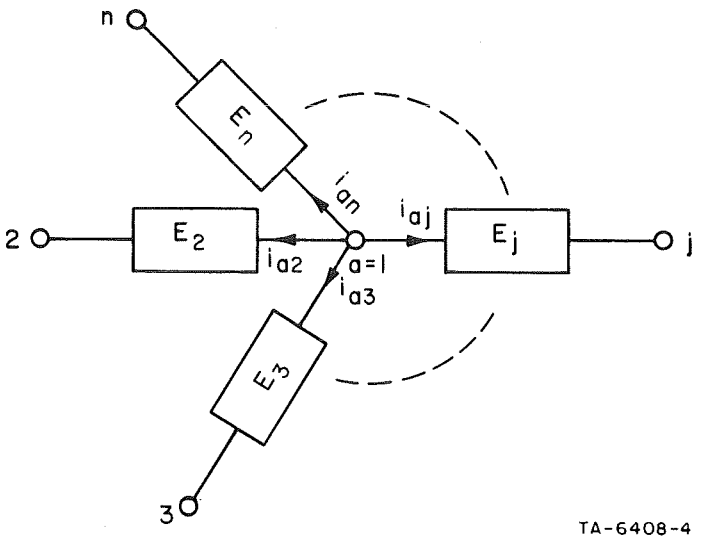


Figure 4 CIRCUIT ELEMENTS BETWEEN NODE a AND OTHER NETWORK NODES

n - 1 currents and summing, we obtain

$$\sum_j i_{aj} = \left(\sum_j \Delta H_j\right) V_a - \sum_j \Delta H_j V_j - \sum_j \Delta T_j \quad .$$

But, following Kirchoff's current law, $\sum_j i_{aj} = 0$. Hence,

$$T_a = H_{aa} V_a + \sum_j H_{aj} V_j \quad , \tag{11}$$

where

$$H_{aa} = \sum_j \Delta H_j \quad , \tag{12}$$

$$H_{aj} = -\Delta H_j \quad , \tag{13}$$

and

$$T_a = \sum_j \Delta T_j \quad . \tag{14}$$

Applying Eq. (11) to each of the n nodes (*i.e.*, letting $a = 1,2,\ldots n$), we obtain the following matrix equation

$$\begin{bmatrix} T_1 \\ T_2 \\ \cdot \\ \cdot \\ \cdot \\ T_j \\ \cdot \\ \cdot \\ \cdot \\ T_n \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1j} & \cdots & H_{1n} \\ H_{21} & H_{22} & \cdots & H_{2j} & \cdots & H_{2n} \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ H_{j1} & H_{j2} & \cdots & H_{jj} & \cdots & H_{jn} \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ H_{n1} & H_{n2} & \cdots & H_{nj} & \cdots & H_{nn} \end{bmatrix} \times \begin{bmatrix} V_1 \\ V_2 \\ \cdot \\ \cdot \\ \cdot \\ V_j \\ \cdot \\ \cdot \\ \cdot \\ V_n \end{bmatrix} \qquad (15)$$

or, in brief,

$$T] = [H] \times V] \qquad (16)$$

b. Filling the $[H]$ and $T]$ Matrices

In Fig. 4 and Eq. (11), the attention was focused on one node (Node a), and the effect of all the circuit elements on this node was examined. In filling the $[H]$ and $T]$ matrices, however, each circuit element is examined separately by computing its contribution to the H and T elements that correspond to its terminals. Initially, the elements of the $[H]$ and $T]$ matrices are set to zero. All the circuit elements of a given type (*e.g.*, sources) are scanned, and the contributions of each of these elements to the matrix elements corresponding to its two nodes (or three nodes in case of a transistor) are computed. This computation procedure is repeated for the remaining circuit-element types. If no circuit element exists between, say, Nodes a and m, then $\Delta H_m = 0$ and, following Eq. (13), $H_{am} = 0$. On the other hand, if Node a is common to ℓ circuit elements, then, following Eqs. (12) and (14), H_{aa} is the sum of ℓ ΔH terms and T_a is the sum of ℓ ΔT terms.

The ΔH and ΔT terms corresponding to each node whose voltage is to be solved will next be examined separately for a two-terminal circuit element and for an active three-terminal circuit element (transistor).

2. A Two-Terminal Circuit Element

Consider a two-terminal circuit element across Nodes a and b, Fig. 1. In filling the $[H]$ and $T]$ matrices, three cases are distinguished, as shown in Fig. 5:

(1) Both V_a and V_b are unknown

(2) V_a is unknown and $V_b = V_{sb}$ is known

(3) $V_a = V_{sa}$ is known and V_b is unknown.

In Case (1), both nodes are floating. In Case (2), Node b is tied to either a voltage source ($V_{sb} \neq 0$) or ground ($V_{sb} = 0$). In Case (3), Node a is tied to either a voltage source ($V_{sa} \neq 0$) or ground ($V_{sa} = 0$). Let us examine each case separately.

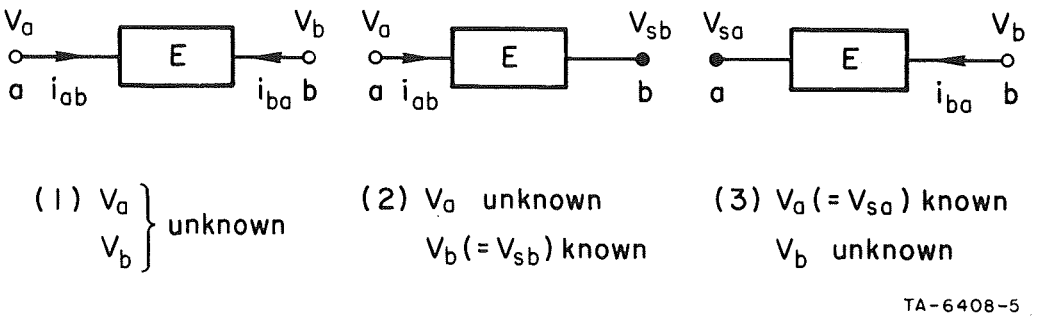


Figure 5 THREE CASES OF NODAL VOLTAGES OF A TWO-TERMINAL CIRCUIT ELEMENT

a. Case (1): V_a And V_b Are Unknown

Let us first consider a current flow from Node a to Node b. Following Eqs. (12), (13), and (14), the contributions of this circuit element to the matrix elements H_{aa}, H_{ab}, and T_a are

$$\Delta H_{aa} = \Delta H \quad , \tag{17}$$

$$\Delta H_{ab} = -\Delta H \quad , \tag{18}$$

and

$$\Delta T_a = \Delta T \quad . \tag{19}$$

Thus, Eq. (1) may be written in the form

$$i_{ab} = \Delta H_{aa} V_a + \Delta H_{ab} V_b - \Delta T_a \quad . \tag{20}$$

So far we have referred to a current flow from Node a to Node b. However, in order to completely fill the matrices, we also have to consider the current i_{ba} from Node b to Node a. Since $i_{ba} = -i_{ab}$, Eq. (1) yields

$$i_{ba} = \Delta H(V_b - V_a) + \Delta T \quad . \tag{21}$$

However, by interchanging a and b in Eq. (20) we also obtain

$$i_{ba} = \Delta H_{bb} V_b + \Delta H_{ba} V_a - \Delta T_b \quad . \tag{22}$$

Equating Eqs. (21) and (22), we find that

$$\Delta H_{bb} = \Delta H \quad , \tag{23}$$

$$\Delta H_{ba} = -\Delta H \quad , \tag{24}$$

and

$$\Delta T_b = -\Delta T \quad . \tag{25}$$

Note that $\Delta H_{aa} = \Delta H_{bb}$ and $\Delta H_{ab} = \Delta H_{ba}$, but $\Delta T_a = -\Delta T_b$.

b. Case (2): V_a Is Unknown; $V_b (= V_{sb})$ Is Known

Substituting $V_b = V_{sb}$ into Eq. (1) gives

$$i_{ab} = \Delta H(V_a - V_{sb}) - \Delta T \quad . \tag{26}$$

Since V_b is known, the matrix equation, Eq. (15), does not include V_b and its associated elements. The values of H_{ab}, H_{bb}, H_{ba}, and T_b are therefore not computed and Eq. (20) is modified to

$$i_{ab} = \Delta H_{aa} V_a - \Delta T_a \quad . \tag{27}$$

By equating Eqs. (26) and (27) we find that

$$\Delta H_{aa} = \Delta H \tag{28}$$

and

$$\Delta T_a = \Delta T + \Delta H \cdot V_{sb} \quad . \tag{29}$$

Case (3): $V_a (= V_{sa})$ Is Known; V_b Is Unknown

Substituting $V_a = V_{sa}$ into Eq. (21) gives

$$i_{ba} = \Delta H (V_b - V_{sa}) + \Delta T \quad . \tag{30}$$

The same arguments for Node b in Case (2) now hold for Node a. Thus, Eq. (22) is modified to

$$i_{ba} = \Delta H_{bb} V_b - \Delta T_b \quad , \tag{31}$$

By equating Eqs. (30) and (31) we find that

$$\Delta H_{bb} = \Delta H \tag{32}$$

and

$$\Delta T_b = -\Delta T + \Delta H \cdot V_{sa} \quad . \tag{33}$$

3. An Active Three-Terminal Circuit Element (Transistor)

A transistor model may be described by a three-terminal circuit element, as was described in Report 5. Let B, C, and E denote the nodes corresponding to the base, collector, and emitter, respectively. Assuming that the voltage of at most one node is tied to a voltage source or ground, four cases are distinguished, as shown in Fig. 6. In Case (1) the voltages V_B, V_C, and V_E of the three nodes are unknown, whereas in Case (2), (3), or (4), the voltage of Node B, C, or E, respectively, is known. Let us examine each case separately.

a. Case (1): V_B, V_C, and V_E Are Unknown

The base current of a transistor is

$$i_b = i_{bc} + i_{be} \quad , \tag{34}$$

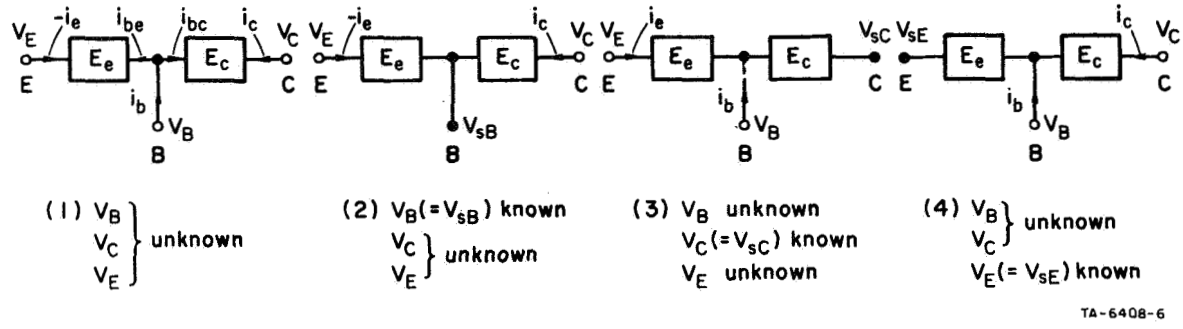


Figure 6 FOUR CASES OF NODAL VOLTAGES OF A THREE-TERMINAL CIRCUIT ELEMENT (Transistor)

where i_{bc} and i_{be} may be expressed as

$$i_{bc} = \Delta H_{(cc)}(V_B - V_C) + \Delta H_{(ce)}(V_B - V_E) - \Delta T_{(c)} \tag{35}$$

and

$$i_{be} = \Delta H_{(ec)}(V_B - V_C) + \Delta H_{(ee)}(V_B - V_E) - \Delta T_{(e)} \tag{36}$$

[Note that the currents of a two-terminal device are described in Eqs. (1) and (21) by using $(2\text{-}1)^2 = 1$ ΔH term and $(2\text{-}1) = 1$ ΔT term, whereas the currents of an active three-terminal device are described above by using $(3\text{-}1)^2 = 4$ ΔH terms and $(3\text{-}1) = 2$ ΔT terms.]

Combining Eqs. (34), (35), and (36), we obtain

$$i_b = \Delta H_{bb}V_B + \Delta H_{bc}V_C + \Delta H_{be}V_E - \Delta T_b \quad , \tag{37}$$

where

$$\Delta H_{bb} = \Delta H_{(cc)} + \Delta H_{(ec)} + \Delta H_{(ee)} + \Delta H_{(ce)} \quad , \tag{38}$$

$$\Delta H_{bc} = -[\Delta H_{(cc)} + \Delta H_{(ec)}] \quad , \tag{39}$$

$$\Delta H_{be} = -[\Delta H_{(ee)} + \Delta H_{(ce)}] \quad , \tag{40}$$

and

$$\Delta T_b = \Delta T_{(c)} + \Delta T_{(e)} \tag{41}$$

Consider now the collector current. Since $i_c = -i_{bc}$, Eq. (35) yields

$$i_c = \Delta H_{cb} V_B + \Delta H_{cc} V_C + \Delta H_{ce} V_E - \Delta T_c \tag{42}$$

where

$$\Delta H_{cb} = -[\Delta H_{(cc)} + \Delta H_{(ce)}] \quad , \tag{43}$$

$$\Delta H_{cc} = \Delta H_{(cc)} \quad , \tag{44}$$

$$\Delta H_{ce} = \Delta H_{(ce)} \quad , \tag{45}$$

and

$$\Delta T_c = -\Delta T_{(c)} \quad . \tag{46}$$

Finally, consider the emitter current. Since $i_e = i_{be}$, Eq. (36) yields

$$-i_e = \Delta H_{eb} V_B + \Delta H_{ec} V_C + \Delta H_{ee} V_E - \Delta T_e \tag{47}$$

where

$$\Delta H_{eb} = -[\Delta H_{(ee)} + \Delta H_{(ec)}] \tag{48}$$

$$\Delta H_{ec} = \Delta H_{(ec)} \tag{49}$$

$$\Delta H_{ee} = \Delta H_{(ee)} \tag{50}$$

and

$$\Delta T_e = -\Delta T_{(e)} \tag{51}$$

A summary of Eqs. (37) through (51) is given as follows:

$$\begin{bmatrix} i_b + \Delta T_b \\ i_c + \Delta T_c \\ -i_e + \Delta T_e \end{bmatrix} = \begin{bmatrix} \Delta H_{bb} & \Delta H_{bc} & \Delta H_{be} \\ \Delta H_{cb} & \Delta H_{cc} & \Delta H_{ce} \\ \Delta H_{eb} & \Delta H_{ec} & \Delta H_{ee} \end{bmatrix} \times \begin{matrix} V_B \\ V_C \\ V_E \end{matrix} \tag{52}$$

where, for r and s designating b, c, or e, the ΔT_r and ΔH_{rs} elements are expressed in the following tabulated equations:

	ΔT_r =	ΔH_{rs} =		
		b	c	e
b	$\Delta T_{(c)} + \Delta T_{(e)}$	$\Delta H_{(cc)} + \Delta H_{(ec)} + \Delta H_{(ee)} + \Delta H_{(ce)}$	$-\Delta H_{(cc)} - \Delta H_{(ec)}$	$-\Delta H_{(ee)} - \Delta H_{(ce)}$
c	$-\Delta T_{(c)}$	$-\Delta H_{(cc)} - \Delta H_{(ce)}$	$\Delta H_{(cc)}$	$\Delta H_{(ce)}$
e	$-\Delta T_{(e)}$	$-\Delta H_{(ee)} - \Delta H_{(ec)}$	$\Delta H_{(ec)}$	$\Delta H_{(ee)}$

(53)

Note that the terms of each column and each row in the ΔH_{rs} matrix add up to zero. The same is true for the ΔT_r column matrix.

b. Case (2): V_B ($=V_{sB}$) Is Known; V_C and V_E Are Unknown

Since $V_B = V_{sB}$, Eq. (42) is modified to

$$i_c = \Delta H_{cc} V_C + \Delta H_{ce} V_E - \Delta T_c \quad , \tag{54}$$

where ΔH_{cc} and ΔH_{ce} are expressed in Eqs. (44) and (45), and

$$\Delta T_c = -\Delta T_{(c)} + [\Delta H_{(cc)} + \Delta H_{(ce)}] V_{sB} \quad . \tag{55}$$

Similarly, Eq. (47) is modified to

$$-i_e = \Delta H_{ec} V_C + \Delta H_{ee} V_E - \Delta T_e \quad , \tag{56}$$

where ΔH_{ec} and ΔH_{ee} are expressed in Eqs. (49) and (50), and

$$\Delta T_e = -\Delta T_{(e)} + [\Delta H_{(ee)} + \Delta H_{(ec)}] V_{sB} \quad . \tag{57}$$

To summarize, then,

$$\begin{bmatrix} i_c + \Delta T_c \\ -i_e + \Delta T_e \end{bmatrix} = \begin{bmatrix} \Delta H_{cc} & \Delta H_{ce} \\ \Delta H_{ec} & \Delta H_{ee} \end{bmatrix} \times \begin{bmatrix} V_C \\ V_E \end{bmatrix} \quad , \tag{58}$$

where, for r and s designating c or e,

$$\Delta T_r = \begin{array}{c|c} & \\ c & -\Delta T_{(c)} + [\Delta H_{(cc)} + \Delta H_{(ce)}]V_{sB} \\ e & -\Delta T_{(e)} + [\Delta H_{(ee)} + \Delta H_{(ec)}]V_{sB} \end{array} \qquad \Delta H_{rs} = \begin{array}{c|cc} & c & e \\ \hline c & \Delta H_{(cc)} & \Delta H_{(ce)} \\ e & \Delta H_{(ec)} & \Delta H_{(ee)} \end{array} \quad . \tag{59}$$

c. Case (3): V_C $(=V_{sC})$ Is Known; V_B and V_E Are Unknown

Since $V_C = V_{sC}$, Eq. (37) is modified to

$$i_b = \Delta H_{bb}V_B + \Delta H_{be}V_E - \Delta T_b \quad , \tag{60}$$

where ΔH_{bb} and ΔH_{be} are expressed in Eqs. (38) and (40), and

$$\Delta T_b = \Delta T_{(c)} + \Delta T_{(e)} + [\Delta H_{(cc)} + \Delta H_{(ec)}]V_{sC} \quad . \tag{61}$$

Similarly, Eq. (47) is modified to

$$-i_e = \Delta H_{eb}V_B + \Delta H_{ee}V_E - \Delta T_e \quad , \tag{62}$$

where ΔH_{eb} and ΔH_{ee} are expressed in Eqs. (48) and (50), and

$$\Delta T_e = -\Delta T_{(e)} - \Delta H_{(ec)}V_{sC} \quad . \tag{63}$$

To summarize,

$$\begin{bmatrix} i_b + \Delta T_b \\ -i_e + \Delta T_e \end{bmatrix} = \begin{bmatrix} \Delta H_{bb} & \Delta H_{be} \\ \Delta H_{eb} & \Delta H_{ee} \end{bmatrix} \times \begin{bmatrix} V_B \\ V_E \end{bmatrix} \quad , \tag{64}$$

where, for r and s designating b or e,

$$\Delta T_r = \begin{array}{c|c} & \\ b & \Delta T_{(c)} + \Delta T_{(e)} + [\Delta H_{(cc)} + \Delta H_{(ec)}]V_{sC} \\ e & -\Delta T_{(e)} - \Delta H_{(ec)}V_{sC} \end{array} \qquad \Delta H_{rs} = \begin{array}{c|cc} & b & e \\ \hline b & \Delta H_{(cc)} + \Delta H_{(ec)} + \Delta H_{(ee)} + \Delta H_{(ce)} & -\Delta H_{(ee)} - \Delta H_{(ce)} \\ e & -\Delta H_{(ee)} - \Delta H_{(ec)} & \Delta H_{(ee)} \end{array} \quad . \tag{65}$$

d. Case (4): V_E ($=V_{sE}$) Is Known; V_B And V_C Are Unknown

Since $V_E = V_{sE}$, Eq. (37) is modified to

$$i_b = \Delta H_{bb} V_B + \Delta H_{bc} V_C - \Delta T_b \quad , \tag{66}$$

where ΔH_{bb} and ΔH_{bc} are expressed in Eqs. (38) and (39), and

$$\Delta T_b = \Delta T_{(c)} + \Delta T_{(e)} + [\Delta H_{(ee)} + \Delta H_{(ce)}] V_{sE} \quad . \tag{67}$$

Similarly, Eq. (42) is modified to

$$i_c = \Delta H_{cb} V_B + \Delta H_{cc} V_C - \Delta T_c \quad , \tag{68}$$

where ΔH_{cb} and ΔH_{cc} are expressed in Eqs. (43) and (44), and

$$\Delta T_c = -\Delta T_{(c)} - \Delta H_{(ce)} V_{sE} \quad . \tag{69}$$

To summarize,

$$\begin{bmatrix} i_b + \Delta T_b \\ i_c + \Delta T_c \end{bmatrix} = \begin{bmatrix} \Delta H_{bb} & \Delta H_{bc} \\ \Delta H_{cb} & \Delta H_{cc} \end{bmatrix} \times \begin{bmatrix} V_B \\ V_C \end{bmatrix} \quad , \tag{70}$$

where, for r and s designating b or c,

$\Delta T_r =$

b	$\Delta T_{(c)} + \Delta T_{(e)} + [\Delta H_{(ee)} + \Delta H_{(ce)}] V_{sE}$
c	$-\Delta T_{(c)} - \Delta H_{(ce)} V_{sE}$

$\Delta H_{rs} =$

	b	c
b	$\Delta H_{(cc)} + \Delta H_{(ec)} + \Delta H_{(ee)} + \Delta H_{(ce)}$	$-\Delta H_{(cc)} - \Delta H_{(ec)}$
c	$-\Delta H_{(cc)} - \Delta H_{(ce)}$	$\Delta H_{(cc)}$

. (71)

B. Calculation of the ΔH and ΔT Terms for Electric Circuit-Element Models

We shall next derive the expression for ΔH and ΔT of various two-terminal electric circuit elements (current sources, resistors, voltage

sources, capacitors, inductors, zener diodes, diodes, and transistors. These expressions will be derived by calculating the current of a two-terminal device and identifying the result with Eq. (1). In the case of a transistor, the expressions for $\Delta H_{(cc)}$, $\Delta H_{(ce)}$, $\Delta T_{(c)}$, $\Delta H_{(ec)}$, $\Delta H_{(ee)}$, and $\Delta T_{(e)}$ will be derived by calculating i_{bc} and i_{be} and identifying the results with Eqs. (35) and (36), respectively. The case of a square-loop core is treated separately in Sec. I-C, pp. 40-78.

1. Current Source

Consider a current source, Fig. 7. Applying Eq. (1) gives

$$i_{ab} = i_s \equiv \Delta H(V_a - V_b) - \Delta T \quad . \tag{72}$$

We thus obtain the simple relations

$$\Delta H = 0 \tag{73}$$

and

$$\Delta T = -i_s \quad . \tag{74}$$

2. Resistor

Consider a resistor, Fig. 8. By inspection,

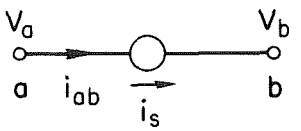
$$i_{ab} = \frac{1}{R}(V_a - V_b) \equiv \Delta H(V_a - V_b) - \Delta T \quad . \tag{75}$$

Hence,

$$\Delta H = \frac{1}{R} \tag{76}$$

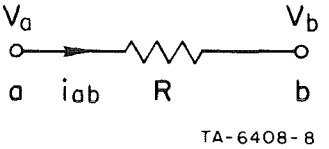
and

$$\Delta T = 0 \quad . \tag{77}$$



TA-6408-7

Figure 7 A CURRENT SOURCE



TA-6408-8

Figure 8 A RESISTOR

Normally, R is constant in time. However, in order to facilitate a switching process during a given period of time, the resistor model may be extended so as to allow R to vary in time. A linear or nonlinear $R(t)$ during the period $t_0 \leq t \leq t_f$ may be described by the parabolic function

$$R(t) = R_0 + (R_f - R_0)\left(\frac{t - t_0}{t_f - t_0}\right)^p \quad ; \tag{78}$$

where R_0 and R_f are the values of R at $t = t_0$ and $t = t_f$, respectively. For either $R_f > R_0$ or $R_f < R_0$, if $p > 1$, then $R(t)$ varies faster near R_f than near R_0 [in the limit, as $p \to \infty$, $R(t) \to R_0$ during $t_0 \leq t < t_f$]; if $p = 1$, then $R(t)$ is linear; and if $p < 1$, then $R(t)$ varies faster near R_0 than near R_f [in the limit, as $p \to 0$, $R(t) \to R_f$ during $t_0 < t \leq t_f$]. These three cases are shown in Fig. 9 for $R_f > R_0$.

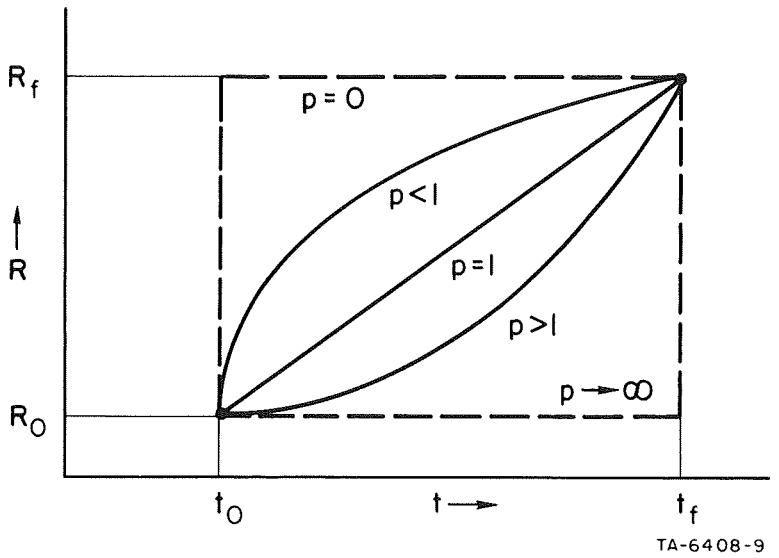


Figure 9 VARIATION OF R(t) WITH DIFFERENT p VALUES

Differentiating Eq. (78) with respect to time, we obtain

$$\dot{R}(t) = (R_f - R_0)p\,\frac{(t - t_0)^{p-1}}{(t_f - t_0)^p} \quad . \tag{79}$$

Thus, at $t = t_0$, $\dot{R} = 0$ if $p > 1$, $\dot{R} = (R_f - R_0)/(t_f - t_0)$ if $p = 1$, and $\dot{R} \to \infty$ if $p < 1$. However, at $t = t_f$, $\dot{R} = p(R_f - R_0)/(t_f - t_0)$ for all cases.

The function expressed by Eq. (78) may be applied with different parameters in successive modes of operation. The values of t_f and R_f of one mode are equal to the values of t_0 and R_0 of the following mode. Designating the mode number by subscript m, we get

$$t_{0(m)} = t_{f(m-1)} \tag{80}$$

and

$$R_{0(m)} = R_{f(m-1)} \quad . \tag{81}$$

If $R(t)$ during a given mode is constant, then $R_f = R_0$ (it is then irrelevant what the value of p is; however, for programming purposes, we shall let $p = 0$). An example is shown in Fig. 10 for variation of $R(t)$ in four modes.

Mode	t_0	R_0	t_f	R_f	p
I	0	R_1	t_1	R_1	0
II	t_1	R_1	t_2	R_2	1
III	t_2	R_2	t_3	R_2	0
IV	t_3	R_2	t_4	0	<1

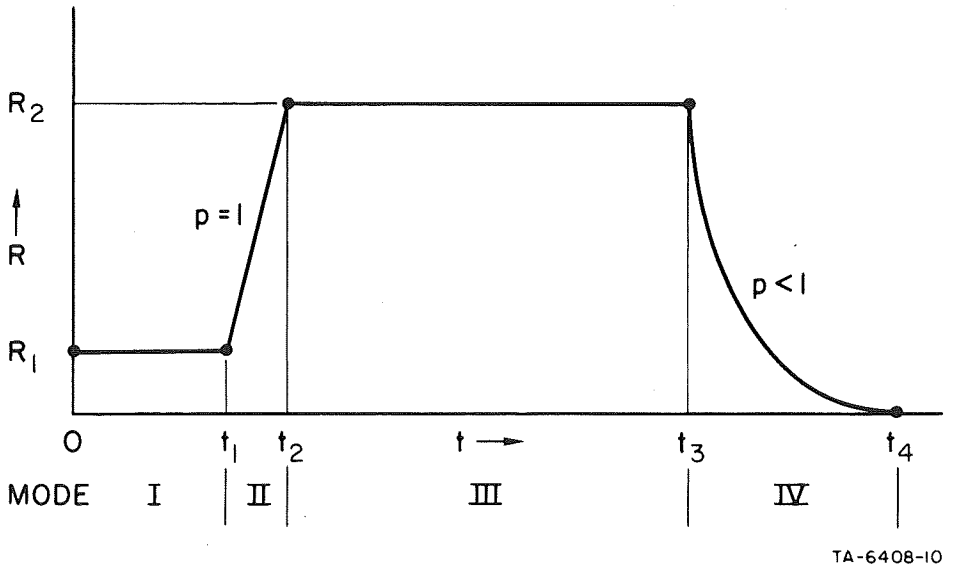


Figure 10 A FOUR-MODE R(t) VARIATION

The required parameters are R_0 and t_0 of the first mode, and R_f, t_f, and p of each of the following modes. Following Eqs. (76) and (77), $\Delta H = 1/R(t)$ and $\Delta T = 0$, where $R(t)$ is computed from Eq. (78).

3. Voltage Source

Consider a voltage source in series with an internal resistance R_s, as shown in Fig. 11.

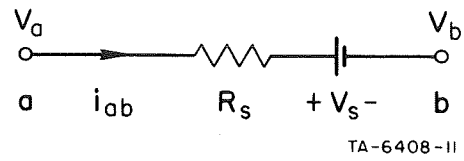


Figure 11 A VOLTAGE SOURCE IN SERIES WITH R_s

Here,

$$i_{ab} = \frac{1}{R_s}\left[(V_a - V_b) - V_s\right] \equiv \Delta H(V_a - V_b) - \Delta T \quad . \tag{82}$$

Hence

$$\Delta H = \frac{1}{R_s} \tag{83}$$

and

$$\Delta T = \frac{V_s}{R_s} \quad . \tag{84}$$

4. Capacitor

Consider a capacitor model consisting of Capacitance C in series with Resistance R_C and in parallel with Leakage Resistance R_{CSH}, as shown in Fig. 12. Let a' be the internal node between R_C and C, and let V'_a be its voltage. The voltage across C is

$$V_C = V'_a - V_b \quad .$$

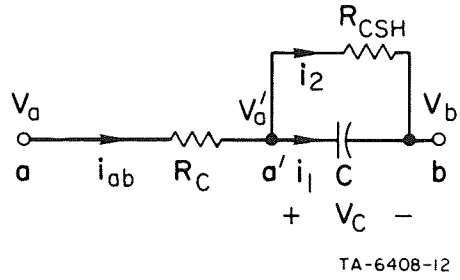


Figure 12 A CAPACITOR MODEL

Applying the trapezoidal integration method, we get

$$V_C \approx V_{C(-1)} + \frac{\Delta t}{2}\left[\dot{V}_{C(-1)} + \dot{V}_C\right] \quad . \tag{85}$$

But, using internal currents i_1 and i_2,

$$\dot{V}_C = \frac{1}{C} i_1 = \frac{1}{C} (i_{ab} - i_2) \quad ,$$

where $i_2 = V_C/R_{CSH}$. Substituting

$$\dot{V}_C = \frac{1}{C} \left[i_{ab} - \frac{V_C}{R_{CSH}} \right]$$

and

$$\dot{V}_{C(-1)} = \frac{1}{C} \left[i_{ab(-1)} - \frac{V_{C(-1)}}{R_{CSH}} \right]$$

into Eq. (85) gives

$$V_C \left(2C + \frac{\Delta t}{R_{CSH}} \right) = V_{C(-1)} \left(2C - \frac{\Delta t}{R_{CSH}} \right) + \Delta t [i_{ab} + i_{ab(-1)}] \quad . \tag{86}$$

Substituting the relations

$$V_C = V_a - V_b - i_{ab} R_C$$

and

$$V_{C(-1)} = V_{a(-1)} - V_{b(-1)} - i_{ab(-1)} R_C$$

into Eq. (86), we obtain

$$i_{ab} = a_2 a_4 (V_a - V_b) - a_3 a_4 \left[V_{a(-1)} - V_{b(-1)} + i_{ab(-1)} \left(\frac{\Delta t}{a_3} - R_C \right) \right] \quad , \tag{87}$$

where

$$a_1 = \Delta t / R_{CSH} \quad , \tag{88}$$

$$a_2 = 2C + a_1 \quad , \tag{89}$$

$$a_3 = 2C - a_1 \quad , \tag{90}$$

and

$$a_4 = 1/(\Delta t + R_C a_2) \quad . \tag{91}$$

Letting

$$h_{aa} = a_2 a_4 \quad , \tag{92}$$

$$b_1 = a_3 a_4 \quad , \tag{93}$$

and

$$b_2 = i_{ab(-1)}\left(\frac{\Delta t}{a_3} - R_C\right) \quad , \tag{94}$$

Eq. (87) is reduced to

$$i_{ab} = h_{aa}(V_a - V_b) - b_1[V_{a(-1)} - V_{b(-1)} + b_2] \quad . \tag{95}$$

Identifying Eqs. (95) and (1) yields

$$\Delta H = h_{aa} \tag{96}$$

and

$$\Delta T = b_1[V_{a(-1)} - V_{b(-1)} + b_2] \quad . \tag{97}$$

5. Inductor

Consider an inductor model of inductance L in series with internal resistance R_L and in parallel with internal shunt resistance R_{LSH}, as shown in Fig. 13. Designating an internal voltage V'_a, the voltage across L is

$$V_L = V'_a - V_b \quad .$$

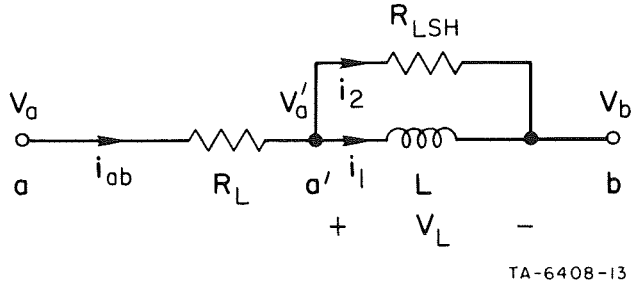


Figure 13 AN INDUCTOR MODEL

Using internal currents i_1 and i_2, the total inductor current is

$$i_{ab} = i_1 + i_2 \quad .$$

By inspection of Fig. 13,

$$\frac{di_1}{dt} = \frac{1}{L} V_L \quad . \tag{98}$$

Applying the trapezoidal integration method, we get

$$i_1 \approx i_{1(-1)} + \Delta t \frac{(di_1/dt)_{(-1)} + (di_1/dt)}{2} \quad . \tag{99}$$

Combining Eqs. (98) and (99) gives

$$i_1 = \left[V_L + V_{L(-1)}\right] \frac{\Delta t}{2L} + i_{1(-1)} \quad .$$

By inspection of Fig. 13,

$$i_2 = V_L/R_{LSH} \quad .$$

Since $i_{ab} = i_1 + i_2$, adding the above two expressions and substituting

$$i_{1(-1)} = i_{ab(-1)} - \frac{V_{L(-1)}}{R_{LSH}}$$

gives

$$i_{ab} = V_L \left(\frac{\Delta t}{2L} + \frac{1}{R_{LSH}}\right) + V_{L(-1)} \left(\frac{\Delta t}{2L} - \frac{1}{R_{LSH}}\right) + i_{ab(-1)} \quad . \tag{100}$$

Substituting

$$V_L = V_a - V_b - i_{ab} R_L$$

and

$$V_{L(-1)} = V_{a(-1)} - V_{b(-1)} - i_{ab(-1)} R_L$$

into Eq. (100), we obtain

$$i_{ab} = h_{aa}(V_a - V_b) - c_1[V_{a(-1)} - V_{b(-1)} + c_2] \quad . \tag{101}$$

where, letting

$$c = 2L/R_{LSH} \tag{102}$$

and

$$b = \frac{1}{2L + R_L(\Delta t + c)} \quad , \tag{103}$$

then

$$h_{aa} = b(c + \Delta t) \quad , \tag{104}$$

$$b_1 = b(c - \Delta t) \quad , \tag{105}$$

and

$$b_2 = \left(\frac{2L}{\Delta t - c} - R_L\right) i_{ab(-1)} \quad . \tag{106}$$

Identifying Eq. (101) with Eq. (1) yields

$$\Delta H = h_{aa} \tag{107}$$

and

$$\Delta T = b_1[V_{a(-1)} - V_{b(-1)} + b_2] \quad . \tag{108}$$

In deriving the above expressions for ΔH and ΔT, nothing has been stated about whether L is constant or not. Referring to Fig. 13, let us now allow L to be either constant or a function of i_1. Using the non-linear inductor model in Report 5, pp. 83-84, we have

$$L = L_0 e^{-|i_1|/I_{con}} \tag{109}$$

where L_0 is the value of L at $i_1 = 0$ and I_{con} is the current constant. Since

$$i_1 = i_{ab} - i_2$$

and

$$i_2 = V_L/R_{LSH} = (V_a - V_b - i_{ab}R_L)/R_{LSH} ,$$

we obtain

$$i_1 = \frac{i_{ab}(R_{LSH} + R_L) - (V_a - V_b)}{R_{LSH}} . \qquad (110)$$

After $V] = T]/[H]$, Eq. (16), is solved for, the value of i_{ab} through the inductor is computed. This computation is based on Eq. (1) if both V_a and V_b are unknown; on Eq. (27) if V_a is unknown and $V_b = V_{sb}$; or on Eq. (31) if $V_a = V_{sa}$ and V_b is unknown. Knowing the values of i_{ab}, V_a, and V_b, then i_1 and L are computed from Eqs. (110) and (109). The value of L is then used to compute ΔH and ΔT from Eqs. (102) through (108).

For economical reasons, $L(i_1)$ will be computed once every Δt, before the first iteration begins.

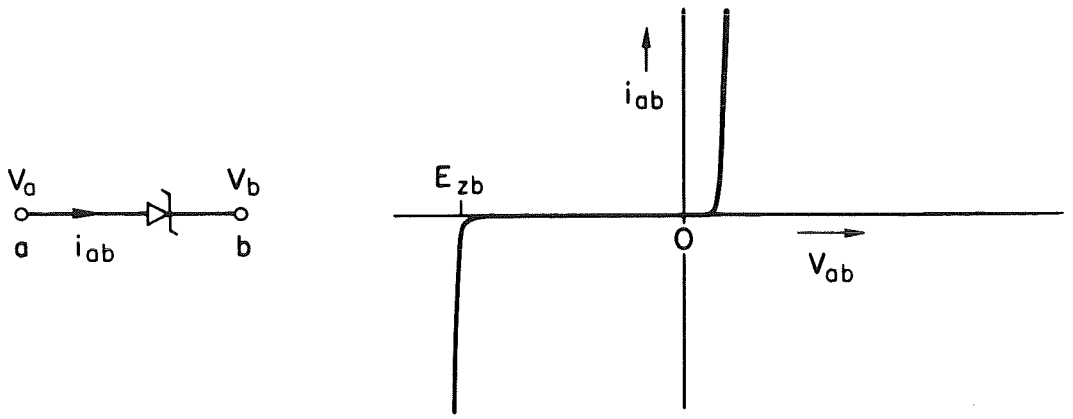
6. Zener Diode

A zener diode and its dc characteristic are shown in Fig. 14(a). An equivalent circuit for the zener diode may be simulated by the model shown in Fig. 14(b). Normally, the zener is reversed biased ($V_{ab} < 0$) and Diode d_2 may be omitted.

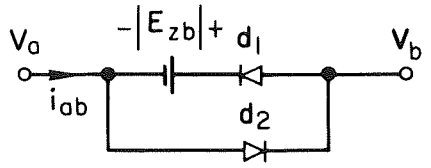
In many cases, a simpler zener-diode model is adequate. If we neglect the zener-diode capacitance and piecewise linearize its dc characteristic, we obtain the simplified model shown in Fig. 14(c). By inspection,

$$i_{ab} = \begin{cases} (V_{ab} - E_{zb})/R_{zb} & \text{for} \quad V_{ab} \leq V_{zb} \qquad (111a) \\ V_{ab}/R_z & \text{for} \quad V_{zb} \leq V_{ab} \leq V_{zf} \qquad (111b) \\ (V_{ab} - E_{zf})/R_{zf} & \text{for} \quad V_{zf} \leq V_{ab} , \qquad (111c) \end{cases}$$

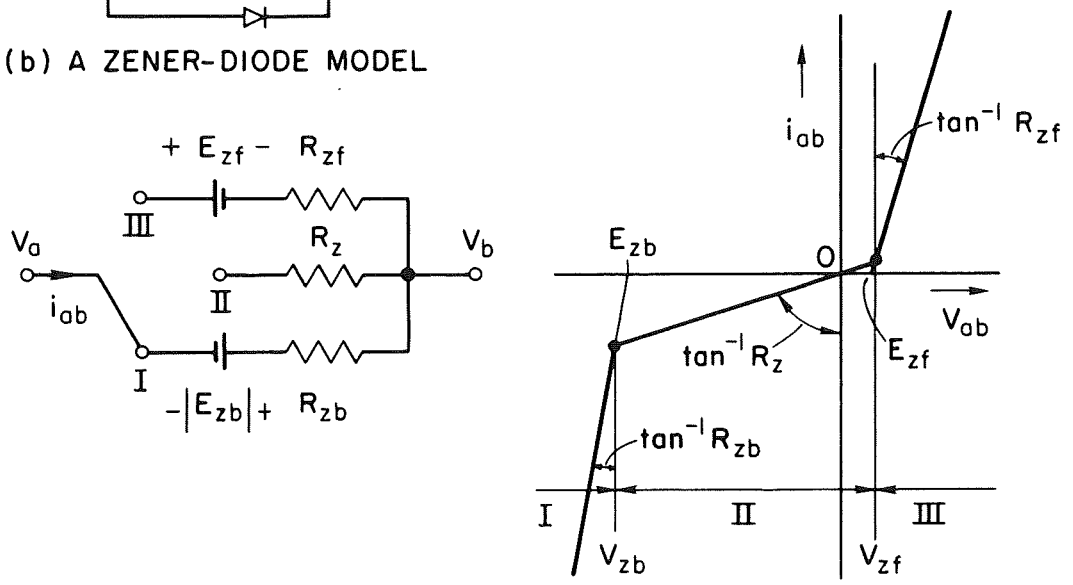
where E_{zb} and E_{zf} are constant voltages, R_{zb}, R_z, and R_{zf} are constant resistances, and V_{zb} and V_{zf} are boundary voltages. Note that E_{zb} and V_{zb} have negative values. By equating V_{ab}/R_z to $(V_{ab} - E_{zb})/R_{zb}$ and to



(a) A ZENER DIODE AND ITS dc CHARACTERISTIC



(b) A ZENER-DIODE MODEL



(c) A PIECEWISE LINEAR ZENER-DIODE MODEL AND ITS CHARACTERISTIC

TA-6408-14

Figure 14 ZENER-DIODE MODELS

$(V_{ab} - E_{zf})/R_{zf}$ we obtain, respectively,

$$V_{zb} = E_{zb} \frac{R_z}{R_z - R_{zb}} \tag{112}$$

and

$$V_{zf} = E_{zf} \frac{R_z}{R_z - R_{zf}} \quad . \tag{113}$$

From Eqs. (111) and the expressions for ΔH and ΔT of a voltage source in series with a resistance, Eq. (83) and (84), we obtain

$$\Delta H = \begin{cases} 1/R_{zb} & \text{for} \quad V_{ab} \leq V_{zb} & (114a) \\ 1/R_z & \text{for} \quad V_{zb} \leq V_{ab} \leq V_{zf} & (114b) \\ 1/R_{zf} & \text{for} \quad V_{zf} \leq V_{ab} & (114c) \end{cases}$$

and

$$\Delta T = \begin{cases} E_{zb}/R_{zb} & \text{for} \quad V_{ab} \leq V_{zb} & (115a) \\ 0 & \text{for} \quad V_{zb} \leq V_{ab} \leq V_{zf} & (115b) \\ E_{zf}/R_{zf} & \text{for} \quad V_{zf} \leq V_{ab} \quad . & (115c) \end{cases}$$

Since a zener diode is seldom forward biased and since the variations in the values of E_{zf} and R_{zf} among the different diodes are negligible, we shall further simplify the zener-diode model by assuming that $E_{zf} = 0.7V$ and $R_{zf} = 1.0\Omega$. Consequently, three parameters are needed:

R_z - high zener-diode resistance,

R_{zb} - low back zener-diode resistance,

and

E_{zb} - extrapolated back zener-diode voltage threshold.

Thus, if $V_z > 0.7V$, then $\Delta H = 1.0\Omega^{-1}$ and $\Delta T = 0.7A$.

7. Diode

A diode model is shown in Fig. 15. This is the same model that was used in Report 5, pp. 85-87, except for a minor difference: The diffusion

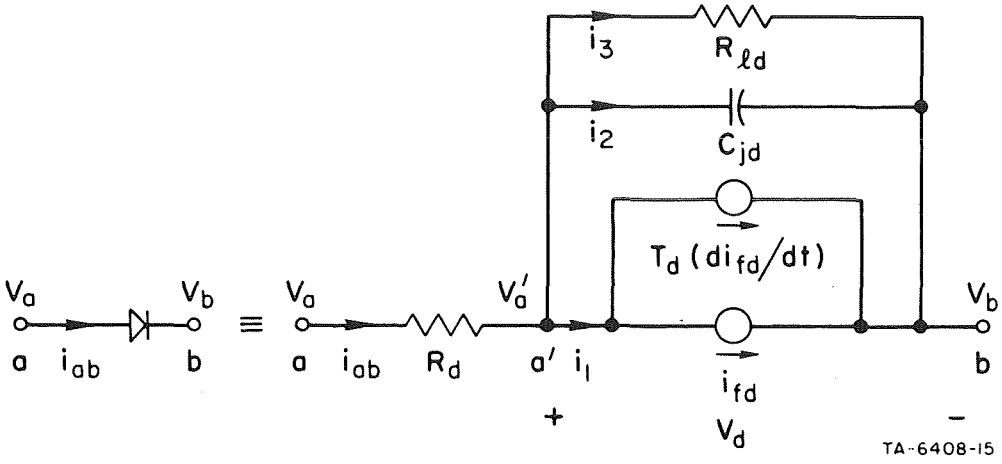


Figure 15 A DIODE MODEL

capacitance, C_{dd}, is replaced by a current source $T_d(di_{fd}/dt)$ whose magnitude is identical with the current through C_{dd}. Accordingly,

$$T_d \frac{di_{fd}}{dt} = T_d \frac{di_{fd}}{dv_d} \frac{dV_d}{dt} \equiv C_{dd} \frac{dV_d}{dt} \quad , \tag{116}$$

where [see Report 5, Eqs. (85), (85a), and (87)]

$$i_{fd} = I_{sd} \left(e^{V_d/\theta_{md}} - 1 \right) \tag{117}$$

and

$$C_{dd} = k_d(i_{fd} + I_{sd}) = k_d I_{sd} e^{V_d/\theta_{md}} \tag{118}$$

and where

$$\theta_{md} = \frac{kT}{q} m_d = 0.86 \cdot 10^{-4} \cdot T \cdot m_d \quad . \tag{119}$$

Transforming Eq. (117) into the form $V_d(i_{fd})$, we obtain

$$V_d = \theta_{md} \ell n \left(1 + \frac{i_{fd}}{I_{sd}} \right) \quad . \tag{120}$$

On the basis of Eqs. (116) through (118), we obtain the relation

$$T_d = k_d \theta_{md} \quad . \tag{121}$$

Instead of using the parameters θ_{md} and k_d, we shall now use the parameters m_d and T_d.

By inspection of Fig. 15,

$$i_{ab} = i_1 + i_2 + i_3 \quad .$$

Let us calculate each of the three current components separately, starting with i_1. By inspection,

$$i_1 = i_{fd} + T_d \frac{di_{fd}}{dt} \approx i_{fd} + T_d \frac{i_{fd} - i_{fd(-1)}}{\Delta t} \quad ;$$

hence

$$i_1 = i_{fd} \left(1 + \frac{T_d}{\Delta t}\right) - \frac{T_d}{\Delta t} i_{fd(-1)} \quad . \tag{122}$$

In order to eventually identify i_{ab} with Eq. (1), we must express i_1 as a linear function of V_d. Since i_{fd} is an exponential function of V_d, Eq. (117), we shall resort to linearization of this function. However, since such linearization is valid only along a small segment around the true values of i_{fd} and V_d, the solution must be *iterative*. The MTRAC program uses a procedure which is based on a modified Newton-Raphson iteration method.[18] This procedure is illustrated in Fig. 16 and is described below.

Let the initial point on the i_{fd} *vs.* V_d curve be denoted by 0^*. A tangent to the i_{fd} *vs.* V_d curve is drawn at Point 0^*, and the diode characteristic is then represented by the straight line d_1 in deriving the ΔH and ΔT terms. Considering the entire circuit, the $[H]$ and $T]$ matrices are filled and $V] = T]/[H]$ is solved for. The value of V_d is determined from the solution of $V]$, and corresponds to Point 1 in Fig. 16. The value of $i_{fd(1)}$ is then compared with $i_{fd}[V_{d(1)}]$, *i.e.*, with the value of i_{fd} obtained by substituting $V_d = V_{d(1)}$ into Eq. (117). In this example, the two values are found to be far apart (*i.e.*, by more than 10 percent); hence, convergence has not been achieved. A horizontal straight line is then drawn from Point 1; this line intersects the i_{fd} *vs.* V_d curve at

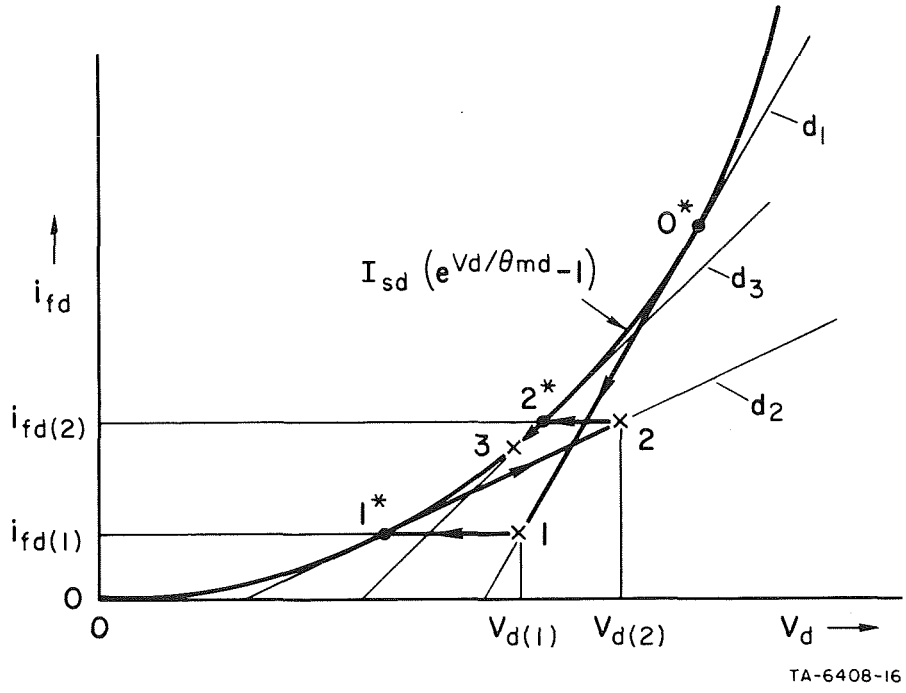


Figure 16 ITERATIVE SOLUTION OF i_{fd} AND V_d

Point 1^*. The above procedure is now repeated for a new linearized diode characteristic, d_2, which is tangent at Point 1^*. The results, marked by Points 2 and 2^*, still do not satisfy the convergence criterion. In the example shown, convergence is finally achieved at the third iteration by reaching Point 3, which is close enough to the i_{fd} *vs.* V_d curve.

On the basis of the procedure described above, the diode characteristic is represented in Fig. 17 by a straight line, d, which is tangent to the i_{fd} *vs.* V_d curve at a point obtained by the end of the previous iteration. The values of i_{fd} and V_d obtained at the previous iteration are designated by the subscript <-1>, *i.e.*, $i_{fd<-1>}$ and $V_{d<-1>}$. The straight line d has a slope s and is described by the function

$$i_{fd} = s(V_d - V_{d<-1>}) + i_{fd<-1>} \quad . \tag{123}$$

Since $s = (di_{fd}/dV_d)\ @\ V_d = V_{d<-1>}$, differentiation of Eq. (117) gives

$$s = \frac{I_{sd}}{\theta_{md}} e^{V_{d<-1>}/\theta_{md}} = \frac{i_{fd<-1>} + I_{sd}}{\theta_{md}} \quad . \tag{124}$$

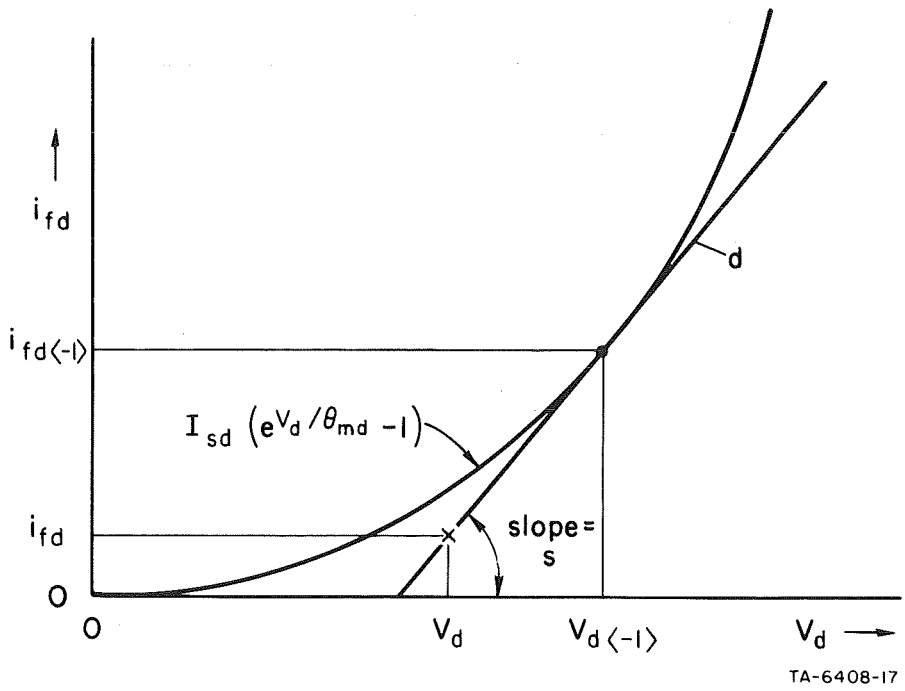


Figure 17 LINEARIZED i_{fd} VERSUS V_d DIODE CHARACTERISTIC

Substituting Eq. (124) into Eq. (123) gives

$$i_{fd} = \frac{i_{fd\langle -1\rangle} + I_{sd}}{\theta_{md}} \left(V_d - V_{d\langle -1\rangle}\right) + i_{fd\langle -1\rangle} \tag{125}$$

Substituting Eq. (125) into Eq. (122) gives

$$i_1 = \left(1 + \frac{T_d}{\Delta t}\right) \frac{i_{fd\langle -1\rangle} + I_{sd}}{\theta_{md}} V_d - \left[\left(1 + \frac{T_d}{\Delta t}\right)\left(\frac{i_{fd\langle -1\rangle} + I_{sd}}{\theta_{md}} V_{d\langle -1\rangle} - i_{fd\langle -1\rangle}\right) + \frac{T_d}{\Delta t}\, i_{fd(-1)}\right] . \tag{126}$$

Referring back to Fig. 15, we now turn to the second and third current components, i_2 and i_3. Applying Eq. (87), in which $R_C = 0$ and $R_{CSH} = \infty$, we get

$$i_2 = \left[V_d - V_{d(-1)}\right] \frac{2C_{jd}}{\Delta t} - i_{2(-1)} .$$

Designating $i_{2(-1)}$ by $i_{C_{jd}(-1)}$, we obtain

$$i_2 = \frac{2C_{jd}}{\Delta t} V_d - \left[\frac{2C_{jd}}{\Delta t} V_{d(-1)} + i_{C_{jd}(-1)}\right] , \qquad (127)$$

where, using Eq. (88) in Report 5,

$$C_{jd} = \begin{cases} \dfrac{C_{j0d}}{[1 - (V_{d<-1>}/V_{\varphi d})]^{N_d}} & \text{for} \quad V_{d<-1>} \leq 0.9V_{\varphi d} \qquad (128a) \\ \dfrac{C_{j0d}}{0.1^{N_d}} & \text{for} \quad V_{d<-1>} > 0.9V_{\varphi d} \quad . \qquad (128b) \end{cases}$$

Note that C_{jd} is limited arbitrarily to a maximum value of $C_{j0d}/(0.1^{N_d})$. This provision is added here in order to prevent an unrealistic blowup of C_{jd} as $V_{d<-1>} \rightarrow V_{\varphi d}$. Thus, for example, if $N_d = 0.5$, then C_{jd} cannot exceed $3.16 \cdot C_{j0d}$.

The third current component is simply

$$i_3 = \frac{1}{R_{\ell_d}} V_d \quad . \qquad (129)$$

Adding i_1, i_2, and i_3, we find that

$$i_{ab} = GV_d - A \quad , \qquad (130)$$

where

$$G = \left(1 + \frac{T_d}{\Delta t}\right) \frac{i_{fd<-1>} + I_{sd}}{\theta_{md}} + \frac{2C_{jd}}{\Delta t} + \frac{1}{R_{\ell_d}} \qquad (131)$$

and

$$A = \left(1 + \frac{T_d}{\Delta t}\right)\left(\frac{i_{fd<-1>} + I_{sd}}{\theta_{md}} V_{d<-1>} - i_{fd<-1>}\right) + \frac{T_d}{\Delta t} i_{fd(-1)} + \frac{2C_{jd}}{\Delta t} V_{d(-1)} + i_{C_{jd}(-1)} \quad . \qquad (132)$$

A slightly simpler version of Eqs. (131) and (132) may be achieved if, instead of using Eq. (127), the approximation

$$i_2 = C_{jd} \frac{V_d - V_{d(-1)}}{\Delta t} \tag{133}$$

is used. In this case,

$$G = \left(1 + \frac{T_d}{\Delta t}\right) \frac{i_{fd<-1>} + I_{sd}}{\theta_{md}} + \frac{C_{jd}}{\Delta t} + \frac{1}{R_{\ell d}} \tag{134}$$

and

$$A = \left(1 + \frac{T_d}{\Delta t}\right)\left(\frac{i_{fd<-1>} + I_{sd}}{\theta_{md}} V_{d<-1>} - i_{fd<-1>}\right) + \frac{T_d}{\Delta t} i_{fd(-1)} + \frac{C_{jd}}{\Delta t} V_{d(-1)} \quad . \tag{135}$$

We now proceed by substituting

$$V_d = V_a - V_b - i_{ab} R_d$$

into Eq. (130) and identifying the resulting expression with Eq. (1). We thus obtain

$$i_{ab} = \frac{G}{1 + GR_d}\left(V_a + V_b\right) - \frac{A}{1 + GR_d}$$

$$\equiv \Delta H(V_a - V_b) - \Delta T \quad . \tag{136}$$

Hence,

$$\Delta H = \frac{G}{1 + GR_d} \tag{137}$$

and

$$\Delta T = \frac{A}{1 + GR_d} \quad . \tag{138}$$

Another simplification may be obtained by assuming that $R_d = 0$. In this case,

$$\Delta H = G \tag{139}$$

and

$$\Delta T = A \quad . \tag{140}$$

Equations (134), (135), (139), and (140) are presently used in the MTRAC program to compute G, A, ΔH, and ΔT, respectively. Let us examine these equations further. The noniterative terms in the expression for G and A, Eqs. (134) and (135), need not be computed every iteration. Such terms are classified into two categories:

(1) Constant terms

$$c_1 = 0.9V_{\varphi d} \tag{141}$$

$$c_2 = 1/\theta_{md} \tag{142}$$

$$c_3 = 1/R_{\ell d} \tag{143}$$

$$c_4 = 1/V_{\varphi d} \tag{144}$$

(2) Δt - dependent terms

$$d_{t1} = T_d/\Delta t \tag{145}$$

$$d_{t2} = 1 + (T_d/\Delta t) = 1 + d_{t1} \tag{146}$$

$$d_{t3} = C_{j0d}/\Delta t \quad . \tag{147}$$

In order to achieve efficiency in the computation of ΔH and ΔT, the constant terms are computed once for the entire run, and new Δt-dependent terms are computed whenever there is a change in Δt. Let

$$b = d_{t2}(i_{fd<-1>} + I_{sd})c_2 + c_3 \quad , \tag{148}$$

$$c = d_{t2}[(i_{fd<-1>} + I_{sd})c_2V_{d<-1>} - i_{fd<-1>}] + d_{t1}i_{fd(-1)} \quad , \tag{149}$$

and

$$d = \frac{d_{t3}}{\sqrt{1 - c_4 \min\,(V_{d<-1>}, c_1)}} \quad . \tag{150}$$

Substituting Eqs. (141) through (150) into Eqs. (134) and (135) then yields

$$\Delta H = b + d \tag{151}$$

and

$$\Delta T = c + d \cdot V_{d(-1)} \quad . \tag{152}$$

8. Transistor

An *npn* transistor model is shown in Fig. 18. This is the same model that was used in Report 5, pp. 87-89, except for a minor difference similar to the one in the diode model: C_{dc} and C_{de} have been replaced by current sources $T_c(di_{fc}/dt)$ and $T_e(di_{fe}/dt)$, respectively, where, following Eq. (121),

$$T_c = k_c \theta_{mc} \tag{153}$$

and

$$T_e = k_e \theta_{me} \quad . \tag{154}$$

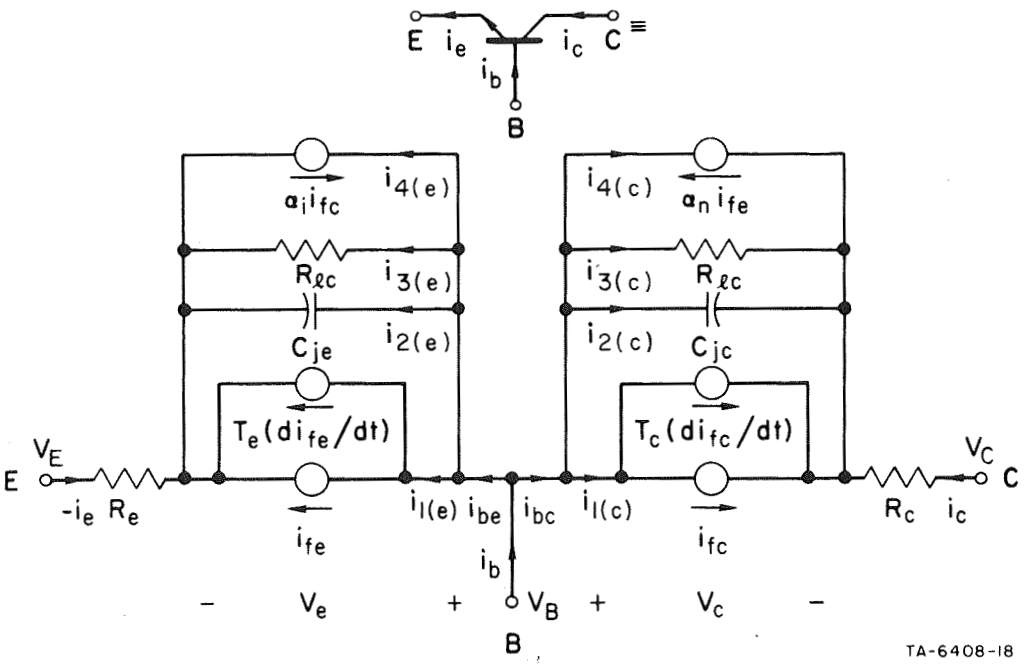


Figure 18 AN npn-TRANSISTOR MODEL

The two *pn* junctions, *B-C* and *B-E*, are each treated as a diode except for the additional current sources, $\alpha_n i_{fe}$ and $\alpha_i i_{fc}$. The convergence routine for the diode, Fig. 16, will also be applied to each of the transistor *p-n* junctions.

Consider first the base-collector current, $i_{bc}(=-i_c)$, which has four components, *i.e.*,

$$i_{bc} = i_{1(c)} + i_{2(c)} + i_{3(c)} + i_{4(c)}$$

The first three components are similar to the diode-current components i_1, i_2, and i_3, respectively. Hence, Eqs. (126), (127), and (129) in which subscript d is replaced by subscript c describe the current components $i_{1(c)}$, $i_{2(c)}$, and $i_{3(c)}$, respectively. The fourth component is

$$i_{4(c)} = -\alpha_n i_{fe}$$

Using modified Eq. (125) in which subscript d is replaced by subscript e in order to express i_{fe}, we obtain

$$i_{4(c)} = -\alpha_n \frac{i_{fe<-1>} + I_{se}}{\theta_{me}} V_e - \alpha_n \left(i_{fe<-1>} - \frac{i_{fe<-1>} + I_{se}}{\theta_{me}} V_{e<-1>} \right) . \tag{155}$$

Adding $i_{1(c)}$, $i_{2(c)}$, $i_{3(c)}$, and $i_{4(c)}$, we find that

$$i_{bc} = G_c V_c + S_c V_e - A_c \tag{156}$$

in which

$$G_c = \left(1 + \frac{T_c}{\Delta t} \right) \frac{i_{fc<-1>} + I_{sc}}{\theta_{mc}} + \frac{2C_{jc}}{\Delta t} + \frac{1}{R\ell_c} \tag{157}$$

$$S_c = -\alpha_n \frac{i_{fe<-1>} + I_{se}}{\theta_{me}} \tag{158}$$

and

$$A_c = \left(1 + \frac{T_c}{\Delta t}\right)\left(\frac{i_{fc<-1>} + I_{sc}}{\theta_{mc}} V_{c<-1>} - i_{fc<-1>}\right) + \frac{T_c}{\Delta t} i_{fc(-1)}$$

$$+ \frac{2C_{jc}}{\Delta t} V_{c(-1)} + i_{C_{jc}(-1)} + \alpha_n \left(i_{fe<-1>} - \frac{i_{fe<-1>} + I_{se}}{\theta_{me}} V_{e<-1>}\right) . \tag{159}$$

If a modified Eq. (133) in which subscript d is replaced by a subscript c is used to describe $i_{2(c)}$, then the expressions for G_c and A_c are slightly simpler:

$$G_c = \left(1 + \frac{T_c}{\Delta t}\right)\frac{i_{fc<-1>} + I_{sc}}{\theta_{mc}} + \frac{C_{jc}}{\Delta t} + \frac{1}{R_{\ell_c}} \tag{160}$$

and

$$A_c = \left(1 + \frac{T_c}{\Delta t}\right)\left(\frac{i_{fc<-1>} + I_{sc}}{\theta_{mc}} V_{c<-1>} - i_{fc<-1>}\right) + \frac{T_c}{\Delta t} i_{fc(-1)}$$

$$+ \frac{C_{jc}}{\Delta t} V_{c(-1)} + \alpha_n \left(i_{fe<-1>} - \frac{i_{fe<-1>} + I_{se}}{\theta_{me}} V_{e<-1>}\right) \tag{161}$$

Consider now the base-emitter current $i_{be}(=i_e)$, Fig. 18. Interchanging subscripts c with e and n with i in Eqs. (156) through (159), we obtain the following relations:

$$i_{be} = G_e V_e + S_e V_c - A_e \tag{162}$$

in which

$$G_e = \left(1 + \frac{T_e}{\Delta t}\right)\frac{i_{fe<-1>} + I_{se}}{\theta_{me}} + \frac{2C_{je}}{\Delta t} + \frac{1}{R_{\ell_e}} , \tag{163}$$

$$S_e = -\alpha_i \frac{i_{fc<-1>} + I_{sc}}{\theta_{mc}} \tag{164}$$

and

$$A_e = \left(1 + \frac{T_e}{\Delta t}\right)\left(\frac{i_{fe<-1>} + I_{se}}{\theta_{me}} V_{e<-1>} - i_{fe<-1>}\right)$$

$$+ \frac{T_e}{\Delta t} i_{fe(-1)} + \frac{2C_{je}}{\Delta t} V_{e(-1)} + i_{C_{je}(-1)}$$

$$+ \alpha_i \left(i_{fc<-1>} - \frac{i_{fc<-1>} + I_{sc}}{\theta_{mc}} V_{c<-1>}\right) \quad . \tag{165}$$

Following Eqs. (160) and (161), a slightly simplified version of G_e and A_e is

$$G_e = \left(1 + \frac{T_e}{\Delta t}\right)\frac{i_{fe<-1>} + I_{se}}{\theta_{me}} + \frac{C_{je}}{\Delta t} + \frac{1}{R_{\ell e}} \tag{166}$$

and

$$A_e = \left(1 + \frac{T_e}{\Delta t}\right)\left(\frac{i_{fe<-1>} + I_{se}}{\theta_{me}} V_{e<-1>} - i_{fe<-1>}\right) + \frac{T_e}{\Delta t} i_{fe(-1)}$$

$$+ \frac{C_{je}}{\Delta t} V_{e(-1)} + \alpha_i \left(i_{fc<-1>} - \frac{i_{fc<-1>} + I_{sc}}{\theta_{mc}} V_{c<-1>}\right) \quad . \tag{167}$$

By inspection of Fig. 18,

$$V_c = V_B - V_C - i_{bc}R_c$$

and

$$V_e = V_B - V_E - i_{be}R_e \quad .$$

Substituting these relations into Eqs. (156) and (162), we obtain

$$i_{bc} = \frac{1}{D}\left\{[G_c(1 + G_eR_e) - S_eS_cR_e](V_B - V_C) + [S_c(1 + G_eR_e) - G_eS_cR_e](V_B - V_E)\right.$$

$$\left. - [A_c(1 + G_eR_e) - A_eS_cR_e]\right\} \tag{168}$$

and

$$i_{be} = \frac{1}{D}\left\{[S_e(1 + G_cR_c) - G_cS_eR_c](V_B - V_C) + [G_e(1 + G_cR_c) - S_cS_eR_c](V_B - V_E) - [A_e(1 + G_cR_c) - A_cS_eR_c]\right\} , \quad (169)$$

where

$$D = (1 + G_cR_c)(1 + G_eR_e) - S_cR_cS_eR_e . \quad (170)$$

By identifying Eq. (168) with Eq. (35) and Eq. (169) with Eq. (36), we finally obtain the ΔH and ΔT terms for an *npn* transistor:

$$\Delta H_{(cc)} = [G_c(1 + G_eR_e) - S_eS_cR_e]/D , \quad (171)$$

$$\Delta H_{(ce)} = [S_c(1 + G_eR_e) - G_eS_cR_e]/D , \quad (172)$$

$$\Delta T_{(c)} = [A_c(1 + G_eR_e) - A_eS_cR_e]/D , \quad (173)$$

$$\Delta H_{(ec)} = [S_e(1 + G_cR_c) - G_cS_eR_c]/D , \quad (174)$$

$$\Delta H_{(ee)} = [G_e(1 + G_cR_c) - S_cS_eR_c]/D , \quad (175)$$

and

$$\Delta T_{(e)} = [A_e(1 + G_cR_c) - A_cS_eR_c]/D . \quad (176)$$

If it is assumed that $R_e = R_c = 0$, then $D = 1$ and Eqs. (171) through (176) are simplified to the following:

$$\Delta H_{(cc)} = G_c , \quad (177)$$

$$\Delta H_{(ce)} = S_c , \quad (178)$$

$$\Delta T_{(c)} = A_c , \quad (179)$$

$$\Delta H_{(ec)} = S_e , \quad (180)$$

$$\Delta H_{(ee)} = G_e \quad , \tag{181}$$

and

$$\Delta T_{(e)} = A_e \quad . \tag{182}$$

The values of $\Delta H_{(cc)}$, $\Delta H_{(ce)}$, $\Delta T_{(c)}$, $\Delta H_{(ec)}$, $\Delta H_{(ee)}$, and $\Delta T_{(e)}$ are utilized to compute the values of the ΔH and ΔT matrix elements following the summarized tabulation in Eqs. (53), (59), (65), and (71) for, respectively, Cases (1), (2), (3), and (4) in Fig. 6.

In the presently used MTRAC program, Eqs. (160), (158), (161), (166), (164), and (167) are used to compute G_c, S_c, A_c, G_e, S_e, and A_e, respectively, and Eqs. (177) through (182) are used to compute the ΔH and ΔT terms. As in the case of a diode, two groups of noniterative terms are distinguished:

(1) Constant terms

$$c_{1c} = 0.9\ V_{\varphi c} \tag{183}$$

$$c_{2c} = 1/\theta_{mc} \tag{184}$$

$$c_{3c} = 1/R\ell_c \tag{185}$$

$$c_{4c} = 1/V_{\varphi c} \tag{186}$$

$$c_{1e} = 0.9\ V_{\varphi e} \tag{187}$$

$$c_{2e} = 1/\theta_{me} \tag{188}$$

$$c_{3e} = 1/R\ell_e \tag{189}$$

$$c_{4e} = 1/V_{\varphi e} \tag{190}$$

$$\alpha_n = \beta_n/(1 + \beta_n) \tag{191}$$

$$\alpha_i = \beta_i/(1 + \beta_i) \tag{192}$$

(2) Δt-dependent terms

$$d_{t1c} = T_c/\Delta t \tag{193}$$

$$d_{t2c} = 1 + (T_c/\Delta t) = 1 + d_{t1c} \tag{194}$$

$$d_{t3c} = C_{j0c}/\Delta t \tag{195}$$

$$d_{t1e} = T_e/\Delta t \tag{196}$$

$$d_{t2e} = 1 + (T_e/\Delta t) = 1 + d_{t1e} \tag{197}$$

$$d_{t3e} = C_{j0e}/\Delta t \quad . \tag{198}$$

Let us define the following relations:

$$b_c = d_{t2c}(i_{fc<-1>} + I_{sc})c_{2c} + c_{3c} \quad , \tag{199}$$

$$c_c = d_{t2c}[(i_{fc<-1>} + I_{sc})c_{2c}V_{c<-1>} - i_{fc<-1>}] + d_{t1c}i_{fc(-1)} \quad , \tag{200}$$

$$d_c = \frac{d_{t3c}}{\sqrt{1 - c_{4c}\min(V_{c<-1>}, c_{1c})}} \quad , \tag{201}$$

$$e_c = \alpha_n[i_{fe<-1>} - (i_{fe<-1>} + I_{se})c_{2e}V_{e<-1>}] \quad , \tag{202}$$

$$b_e = d_{t2e}(i_{fe<-1>} + I_{se})c_{2e} + c_{3e} \quad , \tag{203}$$

$$c_e = d_{t2e}[(i_{fe<-1>} + I_{se})c_{2e}V_{e<-1>} - i_{fe<-1>}] + d_{t1e}i_{fe(-1)} \quad , \tag{204}$$

$$d_e = \frac{d_{t3e}}{\sqrt{1 - c_{4e}\min(V_{e<-1>}, c_{1e})}} \quad , \tag{205}$$

and

$$e_e = \alpha_i[i_{fc<-1>} - (i_{fc<-1>} + I_{sc})c_{2c}V_{c<-1>}] \quad . \tag{206}$$

Substituting these relations into Eqs. (158), (160), (161), (164), (166), (167), and (177) through (182), we obtain the following results:

$$\Delta H_{(cc)} = b_c + d_c \quad , \tag{207}$$

$$\Delta H_{(ce)} = -\alpha_n(i_{fe<-1>} + I_{sc})c_{2e} \quad , \tag{208}$$

$$\Delta T_{(c)} = c_c + d_c V_{c(-1)} + e_c \quad , \tag{209}$$

$$\Delta H_{(ee)} = b_e + d_e \quad , \tag{210}$$

$$\Delta H_{(ec)} = -\alpha_i(i_{fc<-1>} + I_{sc})c_{2c} \quad , \tag{211}$$

and

$$\Delta T_{(e)} = c_e + d_e V_{e(-1)} + e_e \quad . \tag{212}$$

So far we have assumed an *npn* transistor. Let us examine the case of a *pnp* transistor. The model in Fig. 18 is valid for a *pnp* transistor if the directions of all the voltages and all the currents are reversed. With negative values of currents and voltages in Eqs. (156) through (167), we find that the equations derived above for an *npn* transistor hold also for a *pnp* transistor if simple modifications are made in Eqs. (157) through (161) and Eqs. (163) through (167) in order to maintain the *magnitudes* of G_c, S_c, A_c, G_e, S_e, and A_e unchanged: $(i_{fc<-1>} + I_{sc})$ is replaced by $(-i_{fc<-1>} + I_{sc})$, and $(i_{fe<-1>} + I_{se})$ is replaced by $(-i_{fe<-1>} + I_{se})$.

C. Square-Loop Magnetic-Core Model

The square-loop core model that was developed in the past (see Report 5, pp. 81-83) has been modified and extended in order to be applicable to ferrite cores and to slowly switching tape-wound metallic cores with different static ϕ(F) characteristics. Adaptation of this model into the TRAC program is described in this section. We shall first calculate ΔH and ΔT for a single switching core and for identically switching cores in parallel and in series. The extended model will then be presented in a form suitable for efficient computation. Finally, we shall describe the input data for different core types and the computation of the corresponding core parameters.

1. Calculation of ΔH and ΔT

a. A Single Switching Core

Consider a square-loop magnetic core linked by n windings, as shown in Fig. 19. Let the current, the number of turns, and the inherent resistance of the jth winding be denoted by i_j, N_j, and R_{wj}, respectively. Assume that the core is initially at negative remanence $\phi = -\phi_r$. For reference, positive direction of every winding current is shown from bottom to top, *i.e.*, in a direction to generate a positive, counterclockwise MMF. The net MMF is

$$F = \sum_{j=1}^{n} N_j i_j \quad , \tag{213}$$

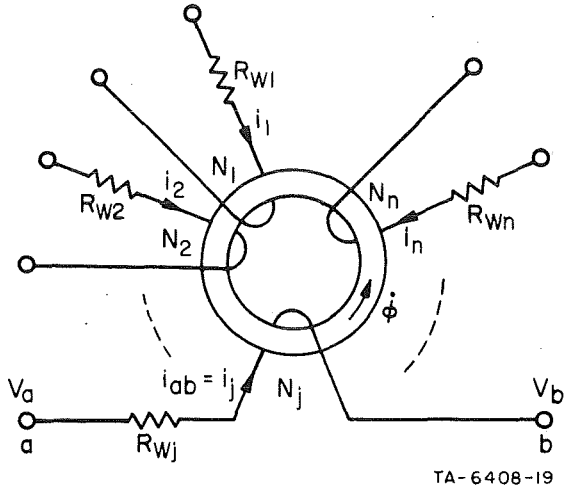


Figure 19 A SQUARE-LOOP MAGNETIC CORE LINKED BY n WINDINGS

where each i_j may be positive or negative. The resulting $\dot{\phi}$ may be positive (as shown in Fig. 19 for reference) or negative, depending on ϕ, F, and $\dot{F}$.

Due to the nonlinearity of the core model, the solution for $\dot{\phi}$ ϕ, and F must be iterative with some means of convergence. Since $\dot{\phi}$ is a function of F, it is also a function of every winding current, *i.e.*, $\dot{\phi} = \dot{\phi}(i_1, \ldots, i_j, \ldots, i_n)$. For a small change in each current between two successive iterations, the first two terms of the Taylor series yield

$$\dot{\phi} = \dot{\phi}_{<-1>} + \sum_{j=1}^{n} (i_j - i_{j<-1>}) \left(\frac{\partial \dot{\phi}}{\partial i_j}\right)_{<-1>} , \qquad (214)$$

where subscript <-1> designates a previous-iteration value.

In view of the relation expressed by Eq. (213) and the fact that $\dot{\phi}$ is common to all the core windings, each jth winding may be treated as a separate two-terminal circuit element for which, following Eq. (214),

$$\dot{\phi} = \dot{\phi}_{<-1>} + (i_j - i_{j<-1>}) \left(\frac{\partial \dot{\phi}}{\partial i_j}\right)_{<-1>} . \qquad (215)$$

Equation (215) is consistent with the Newton-Raphson convergence method for finding the root i_j of the function $f(i_j) = \dot{\phi} - \dot{\phi}_{(solution)} = 0$ because this method is based on equating $f(i_j)$ to the first two terms of its Taylor's series.

Referring to the jth winding in Fig. 19, we obtain

$$V_a - V_b = i_j R_{wj} + N_j \dot{\phi} \quad ;$$

hence,

$$\dot{\phi} = \frac{V_a - V_b - i_j R_{wj}}{N_j} \quad . \tag{216}$$

Now,

$$\frac{\partial \dot{\phi}}{\partial i_j} = \frac{\partial \dot{\phi}}{\partial F} \frac{\partial F}{\partial i_j} \quad ,$$

and following Eq. (213), $\partial F / \partial i_j = N_j$; hence,

$$\frac{\partial \dot{\phi}}{\partial i_j} = N_j \dot{\phi}' \quad , \tag{217}$$

where $\dot{\phi}' = \partial \dot{\phi} / \partial F$. Substituting Eqs. (216) and (217) into Eq. (215) and letting $i_j = i_{ab}$, $i_{j<-1>} = i_{ab<-1>}$, $R_{wj} = R_w$, and $N_j = N$, we obtain the relation

$$i_{ab} = G_c (V_a - V_b - V_c) \quad , \tag{218}$$

where

$$G_c = \frac{1}{N^2 \dot{\phi}'_{<-1>} + R_w} \tag{219}$$

and

$$V_c = N \dot{\phi}_{<-1>} - N^2 \dot{\phi}'_{<-1>} i_{ab<-1>} \quad . \tag{220}$$

Note that $N^2\dot{\phi}'$ represents the core incremental "switching resistance" as seen across a winding of N turns. Computation of $\dot{\phi}$ and $\dot{\phi}'$ is described in Sec. C-2.

Identifying Eq. (218) with Eq. (1) yields the contribution of each core winding to the H and T matrix arrays:

$$\Delta H = G_c \tag{221}$$

and

$$\Delta T = G_c V_c \quad . \tag{222}$$

b. Identically Switching Cores

So far we have treated the usual case of a core whose flux-switching conditions may be different from those of other cores in the circuit. In certain special cases (such as a shift register full of ONEs or ZEROs), there are several cores in parallel or in series that switch identically. Under either of these conditions, the computation may become considerably more efficient if these cores are represented by an equivalent single core. This is especially true for identically switching cores in series because the total number of floating nodes in the entire equivalent circuit is thereby decreased.

Consider first n_p identical cores with identical windings that are connected in parallel, as shown in Fig. 20. If the values of the initial flux and any additional MMF drive are the same for every core, then the cores will switch identically; *i.e.*, the variations of $F(t)$ and $\varphi(t)$ of these cores will be exactly the same. Following Eqs. (5), (6), and (219) through (222), these cores may be represented by an equivalent single core whose ΔH and ΔT are

$$\Delta H_{(p,i)} = n_p G_c \tag{223}$$

and

$$\Delta T_{(p,i)} = n_p G_c V_c \quad . \tag{224}$$

Now consider n_s cores that switch identically in series, as shown in Fig. 21. Following Eqs. (8), (9), and (219) through (222), these cores may be represented by an equivalent single core (across Terminals a and b) whose ΔH and ΔT terms are simply

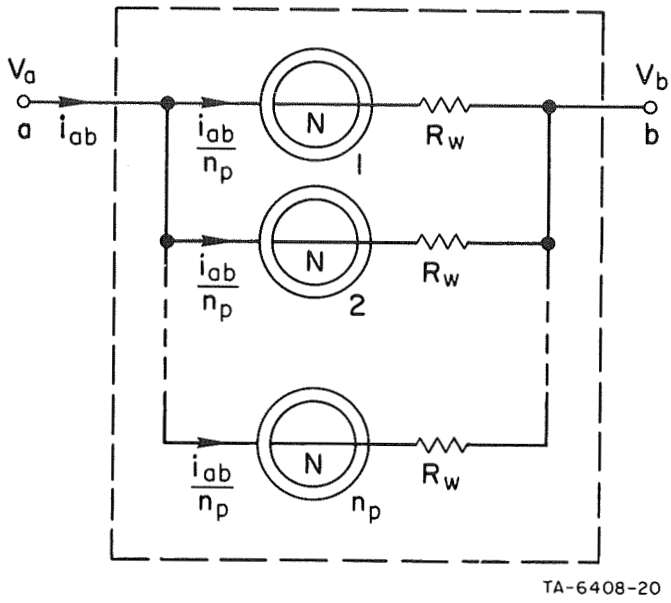


Figure 20 IDENTICALLY SWITCHING CORES IN PARALLEL

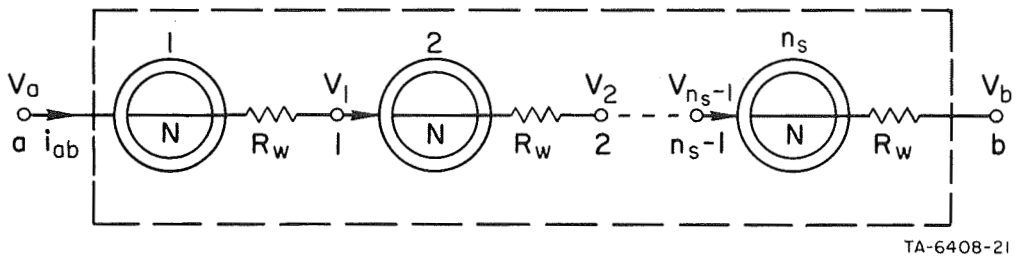


Figure 21 IDENTICALLY SWITCHING CORES IN SERIES

$$\Delta H_{(s,i)} = G_c/n_s \qquad (225)$$

and

$$\Delta T_{(s,i)} = G_c V_c \quad . \qquad (226)$$

2. Square-Loop Core Model

The square-loop magnetic-core model summarized in Report 5, pp. 81-83, will be used with certain modifications that include more general static $\phi(F)$ curve and $\dot{\phi}_p(F)$ curve. We shall maintain the relationship

$$\dot{\phi} = \dot{\phi}_\epsilon + \dot{\phi}_i \quad , \qquad (227)$$

where $\dot{\phi}_\epsilon = \dot{\phi}_\epsilon(F,\dot{F})$ and $\dot{\phi}_i = \dot{\phi}_i(F,\phi)$ are the elastic and inelastic $\dot{\phi}$ components, respectively; hence,

$$\dot{\phi}' = \dot{\phi}'_\epsilon + \dot{\phi}'_i \quad , \qquad (228)$$

where $\dot{\phi}' = \partial\dot{\phi}/\partial F$, $\dot{\phi}'_\epsilon = \partial\dot{\phi}_\epsilon/\partial F$, and $\dot{\phi}'_i = \partial\dot{\phi}_i/\partial F$. Computation of the components of $\dot{\phi}$ and $\dot{\phi}'$ is described next.

a. Elastic $\dot{\phi}$ Component

Computation of $\dot{\phi}_\epsilon$ is based on the relation

$$\dot{\phi}_\epsilon = \epsilon\dot{F} \quad . \qquad (229)$$

Following Eq. (79a) in Report 5,

$$\epsilon = p_1\left[|F|\left(\frac{1}{|F| + p_2} - \frac{1}{|F| + p_3}\right) + \ell n\left(\frac{|F| + p_2}{|F| + p_3}\right)\right] \quad , \qquad (230)$$

where

$$p_1 = \frac{\phi_s - \phi_r}{(l_o - l_i)H_a} \quad , \qquad (231)$$

$$p_2 = H_a l_o \quad , \qquad (232)$$

and

$$p_3 = H_a l_i \quad . \tag{233}$$

In Reports 3 through 5, Eq. (229) was approximated by

$$\dot{\phi}_\epsilon \approx \epsilon \frac{F - F_{(-1)}}{\Delta t} \quad , \tag{234}$$

so that

$$\dot{\phi}'_\epsilon \approx \frac{\epsilon + \epsilon' [F - F_{(-1)}]}{\Delta t} \quad , \tag{235}$$

where $\epsilon' = d\epsilon(F)/dF$. By differentiating Eq. (230) with respect to F, we obtain

$$\epsilon'(F) = sp_1 \left(\frac{1}{|F| + p_2} - \frac{1}{|F| + p_3} \right) \left[2 - |F| \left(\frac{1}{|F| + p_2} + \frac{1}{|F| + p_3} \right) \right] \quad , \tag{236}$$

where $s \equiv \text{sign}(F)$.

For a given $F_{(-1)}$ value, an error in F may cause an excessive error in $\dot{\phi}_\epsilon$ if Δt is small enough. The error in F may stem from a round-off error in the computer and from the maximum relative and absolute errors of winding currents allowed by the user in order to reduce the number of iterations for economical reasons. In order to prevent such undesired and unnatural behavior, fluctuation of $\dot{\phi}_\epsilon(t)$ should be smoothed. The error in the total $\dot{\phi}$ introduced by such a provision is negligible because usually $|\dot{\phi}_\epsilon| << |\dot{\phi}_i|$. Smoothing of $\dot{\phi}_\epsilon(t)$ is based on fitting the previous three $F(t)$ points with a straight line, using the method of least-mean-squared error. The slope of the fitted line is then identified with $\dot{F}$ in Eq. (229) for the purpose of computing $\dot{\phi}_\epsilon$. The linear fitting is formulated as follows.

Suppose that n $(t_i; F_i)$ points are to be fitted by the straight line

$$F(t) = kt + b \quad . \tag{237}$$

The sum of square error is

$$\sum_{i=1}^{n} E_i^2 = \sum_{i=1}^{n} (k t_i + b - F_i)^2 \quad .$$

Differentiating $\sum_{i=1}^{n} E_i^2$ with respect to k and b and equating the derivatives to zero in order to minimize $\sum_{i=1}^{n} E_i^2$, we obtain

$$k = \frac{\sum_{i=1}^{n} F_i \sum_{i=1}^{n} t_i - n \sum_{i=1}^{n} F_i t_i}{\left(\sum_{i=1}^{n} t_i\right)^2 - n \sum_{i=1}^{n} t_i^2} \tag{238}$$

and

$$b = \frac{1}{n}\left(\sum_{i=1}^{n} F_i - k \sum_{i=1}^{n} t_i\right) \quad . \tag{239}$$

Let us use $n = 3$ and refer to the current time as $t = 0$. In Fig. 22 is shown a linear fitting of the three previous points, $[t_{(-1)}; F_{(-1)}]$, $[t_{(-2)}; F_{(-2)}]$, and $[t_{(-3)}; F_{(-3)}]$, where

$$t_{(-1)} = -\Delta t \quad , \tag{240}$$

$$t_{(-2)} = -\Delta t - \Delta t_{(-1)} \quad , \tag{241}$$

$$t_{(-3)} = -\Delta t - \Delta t_{(-1)} - \Delta t_{(-2)} \quad . \tag{242}$$

With $n = 3$, Eqs. (238) and (239) become

$$k = \frac{\sum_{i=1}^{3} F_{(-i)} \sum_{i=1}^{3} t_{(-i)} - 3 \sum_{i=1}^{3} F_{(-i)} t_{(-i)}}{\left[\sum_{i=1}^{3} t_{(-i)}\right]^2 - 3 \sum_{i=1}^{3} t_{(-i)}^2} \tag{243}$$

and

$$b = \frac{1}{3}\left[\sum_{i=1}^{3} F_{(-i)} - k \sum_{i=1}^{3} t_{(-i)}\right] \quad . \tag{244}$$

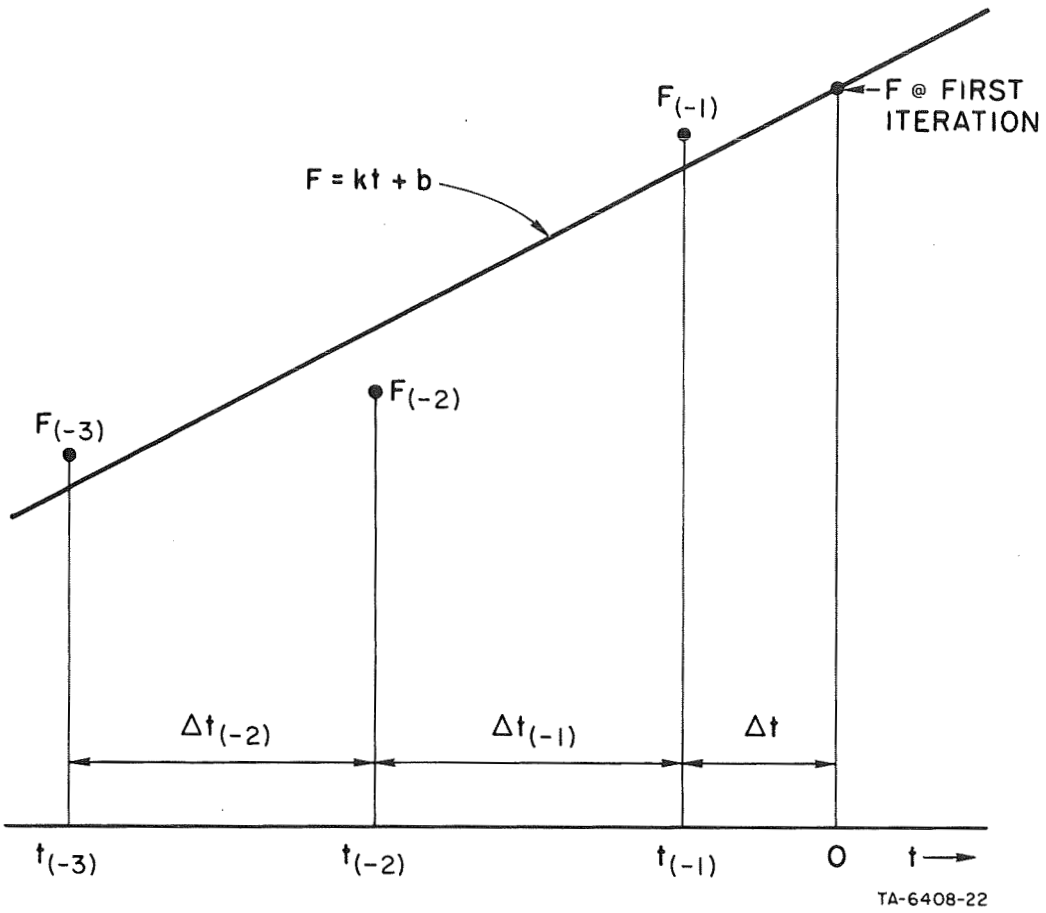


Figure 22 LINEAR FITTING OF THREE F(t) POINTS

Let us use the following abbreviations:

$$S_t = \sum_{i=1}^{3} t_{(-i)} \quad , \tag{245}$$

$$S_{tt} = \sum_{i=1}^{3} t^2_{(-i)} \quad , \tag{246}$$

$$S_{tsq} = S_t^2 - 3S_{tt} \quad , \tag{247}$$

$$S_F = \sum_{i=1}^{3} F_{(-i)} \quad , \tag{248}$$

$$S_{Ft} = \sum_{i=1}^{3} F_{(-i)} t_{(-i)} \quad , \tag{249}$$

and

$$S_{Ftn} = S_F S_t - 3S_{Ft} \quad . \tag{250}$$

Furthermore, one can see from Eq. (237) that k is the average slope of $F(t)$, $\bar{\dot{F}}$, and that, since $t = 0$, b is the guessed value of F for the first iteration. Thus, Eqs. (243) and (244) are reduced to

$$\bar{\dot{F}} = S_{Ftn}/S_{tsq} \tag{251}$$

and, for the first iteration,

$$F = (S_F - \bar{\dot{F}} S_t)/3 \quad . \tag{252}$$

The value of the elastic $\dot{\phi}$ is, then,

$$\dot{\phi}_\epsilon = \epsilon \bar{\dot{F}} \quad . \tag{253}$$

Differentiating with respect to F, we obtain

$$\dot{\phi}'_\epsilon = \epsilon' \bar{\dot{F}} - \epsilon \frac{3}{S_t} \quad . \tag{254}$$

Note that S_t has a negative value.

An alternative method to prevent $\dot{\phi}_\epsilon$ from diverging is to limit the variation of $\dot{\phi}_\epsilon$ relative to $\dot{\phi}_{\epsilon(-1)}$, i.e.,

$$\dot{\phi}_{\epsilon(-1)} - \delta\dot{\phi}_\epsilon \leq \dot{\phi}_\epsilon \leq \dot{\phi}_{\epsilon(-1)} + \delta\dot{\phi}_\epsilon \quad , \tag{255}$$

where

$$\delta\dot{\phi}_\epsilon = \alpha\dot{\phi}_{\epsilon(-1)} + \delta\dot{\phi}_{\epsilon 0} \quad . \tag{256}$$

The positive term $\delta\dot{\phi}_{\epsilon 0}$ is added to allow $\dot{\phi}_\epsilon(t)$ to change sign if $1 \geq \alpha \geq 0$. We shall arbitrarily assume that $\alpha = 1$ and that $\delta\phi_{\epsilon 0} = ZV$, where ZV is a small fraction of $\bar{\dot{\phi}}_i$. Assuming that ZV is ½ percent of an average $\dot{\phi}$ with excess MMF of $F - F''_0 = 0.3\ F''_0$ (cf. Eq. (90) in Report 4), we set

$$ZV = 0.003\,\lambda\,(0.3F_0'')^{\nu} \quad . \tag{257}$$

Thus,

$$\delta\dot{\phi}_{\epsilon} = \dot{\phi}_{\epsilon(-1)} + ZV \quad . \tag{258}$$

Both of the above methods for prevention of excessive $\dot{\phi}_{\epsilon}$ fluctuation will be employed. We shall first compute $\dot{\phi}_{\epsilon}$ from Eq. (253) and then, if necessary, limit $\dot{\phi}_{\epsilon}$ according to Eqs. (255), (257), and (258).

Determination of $\dot{\phi}_{\epsilon}$ and $\dot{\phi}'_{\epsilon}$ using Eqs. (253) and (254) requires the previous three values of F and t. Consequently, computation of $\dot{\phi}_{\epsilon}$ and $\dot{\phi}'_{\epsilon}$ during the first three time steps will be iterative, using Eq. (234) and (235), respectively. Beyond this point, $\dot{\phi}_{\epsilon}$ and $\dot{\phi}'_{\epsilon}$ will be computed once every time step from Eqs. (245) through (258).

b. Inelastic $\dot{\phi}$ Component

A model for the inelastic component, $\dot{\phi}_i(F,\phi)$, was given in Report 5, pp. 81-83, assuming a positive F. However, due to the anti-symmetry of the $\phi(F)$ loop,

$$\dot{\phi}_i(F,\phi) = -\dot{\phi}_i(-F,-\phi) \quad . \tag{259}$$

Hence, computation of a negative $\dot{\phi}_i$ due to a negative F may be performed by first reversing the signs of F and ϕ, then computing $\dot{\phi}_i$ from the core model for positive F, and finally reversing the sign of $\dot{\phi}_i$. In general, therefore, $\dot{\phi}_i$ due to F of either sign will be determined from the relation

$$\dot{\phi}_i(F,\phi) = s\dot{\phi}_i(sF, s\phi) \quad , \tag{260}$$

where

$$s \equiv \text{sign}\,(F) \quad . \tag{261}$$

Note that $sF \geq 0$, but $s\phi$ may be positive, zero, or negative. Combining Eq. (260) and Eqs. (80) and (81) in Report 5, we obtain

$$\dot{\phi}_i = s\dot{\phi}_p \left[1 - \left(\frac{2s\phi + \phi_s - \phi_d}{\phi_s + \phi_d}\right)^2\right] \quad , \tag{262}$$

where $\dot{\phi}_p = \dot{\phi}_p(F)$ and $\phi_d = \phi_d(F)$. Since the computation of $\dot{\phi}_i$ will be performed for $F \geq 0$, we replace $s\dot{\phi}_p$ by $\dot{\phi}_p$ in Eq. (262) and differentiate with respect to F; we thus obtain

$$\dot{\phi}_i' = \left[1 - \left(\frac{2s\phi + \phi_s - \phi_d}{\phi_s + \phi_d}\right)^2\right]\dot{\phi}_p' + 4\dot{\phi}_p \frac{(2s\phi + \phi_s - \phi_d)(s\phi + \phi_s)}{(\phi_s + \phi_d)^3}\phi_d' \quad , \tag{263}$$

where $\dot{\phi}_p' = d\dot{\phi}_p(F)/dF$ and $\phi_d' = d\phi_d(F)/dF$.

The models for $\phi_d(F)$ and $\dot{\phi}_p(F)$ in Report 5 have been extended to include tape-wound cores, in addition to ferrite cores. These models and the corresponding expressions for $\phi_d'(F)$ and $\dot{\phi}_p'(F)$ are described next.

i. Static $\phi(F)$ Models

General Model - A general model for the major static $\phi(F)$ curve, $\phi_d(F)$, is shown in Fig. 23(a). If initially $\phi = -\phi_r$, then $\phi_d(F)$ is divided into five regions. Regions 1, 4, and 5 were used in Reports 2 through 5 to describe $\phi_d(F)$ of ferrite cores with no ϕ_d jump at the threshold MMF $F = F_{d1}$. The addition of Regions 2 and 3 has extended the application of $\phi_d(F)$ to ferrite cores with ϕ_d jumps and to tape-wound metallic cores of different types. If initially $\phi = \phi_r$, then a positive F will change ϕ_d elastically along Region 6 as it approaches ϕ_s. Let us examine each region separately.

Region 1 is an extension of the region of negative saturation. Referring to Eqs. (82) in Report 5, it is described by the function

$$\phi_d = \frac{\phi_s - \phi_r}{(l_o - l_i)H_a} F \ln\left(\frac{F - H_a l_o}{F - H_a l_i}\right) - \phi_r \quad , \tag{264}$$

where l_i and l_o are inside and outside circumferential lengths and H_a is a material saturation constant.

Region 2 is added due to a possible curvature in $\phi_d(F)$, such as may be found in tape-wound cores. This curvature is described by a four-parameter function of the form $p_5 + p_6(F - F_{d1}) + p_8(F - F_{d1})^{p_7}$, which satisfies the following requirements: It is tangent to $\phi_d(F)$ of Region 1 at the point (F_{d1}, ϕ_{d1}), thus fixing p_5 and p_6, and it passes

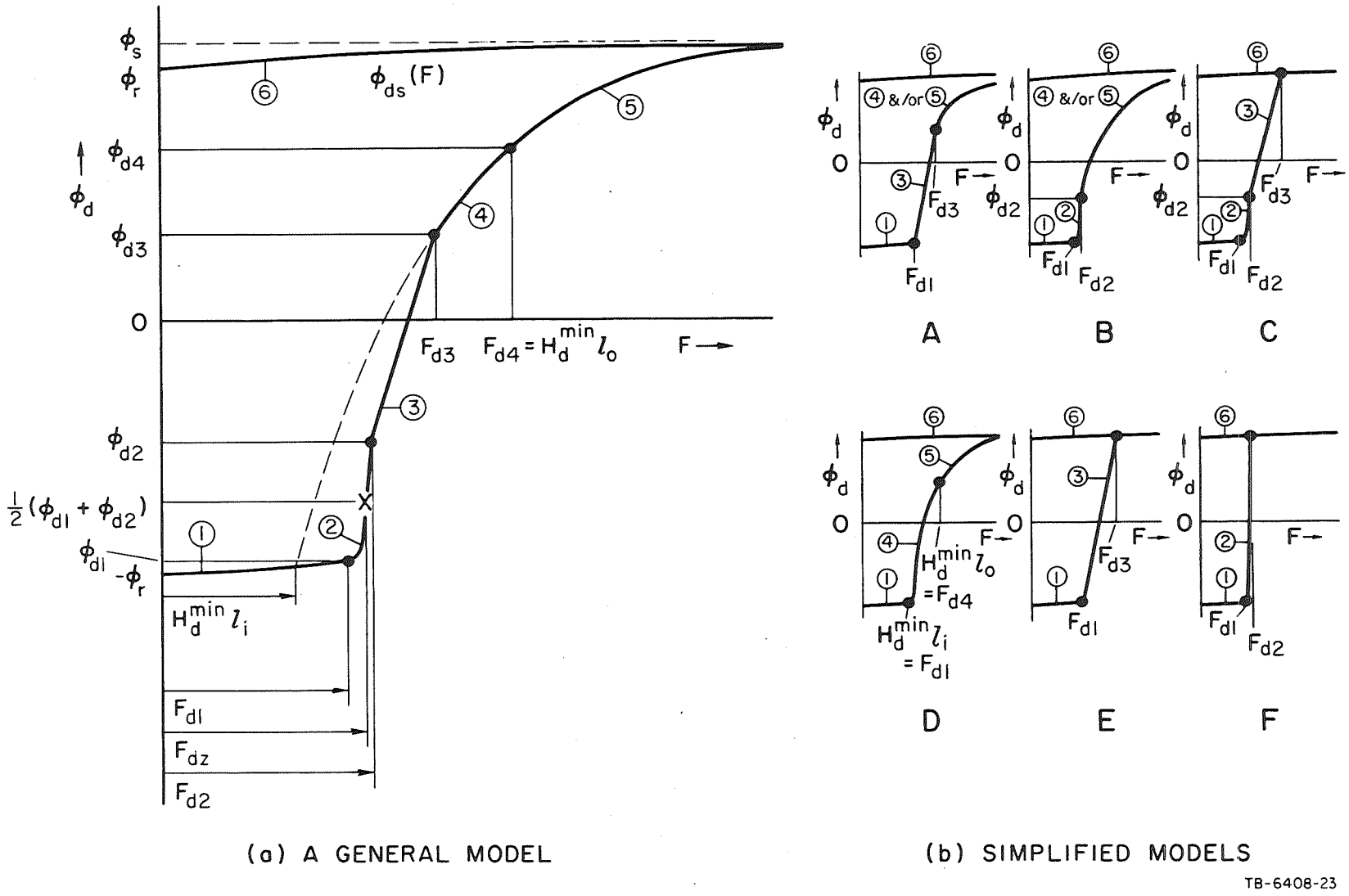


Figure 23 MODELS FOR THE STATIC ϕ(F) CURVE

through the points (F_{d2}, ϕ_{d2}) and $[F_{dz}, \frac{1}{2}(\phi_{d1} + \phi_{d2})]$, where $\frac{1}{2}(F_{d1} + F_{d2}) \leq F_{dz} \leq F_{d2}$, thus fixing p_7 and p_8. Note that F_{dz} affects the curvature of $\phi_d(F)$ and that if $F_{dz} = \frac{1}{2}(F_{d1} + F_{d2})$, then $\phi_d(F)$ becomes a straight line connecting the points (F_{d1}, ϕ_{d1}) and (F_{d2}, ϕ_{d2}). Region 2 may, alternatively, be used to approximate a jump in $\phi_d(F)$ at $F = F_{d1}$ by a monotonically increasing $\phi_d(F)$ (usually, a straight line of a high slope) in order to prevent computational oscillations.

Region 3 is a straight line between the points (F_{d2}, ϕ_{d2}) and (F_{d3}, ϕ_{d3}). This region is added because a large portion of the static $\phi(F)$ curve of a tape-wound core, whose static $B(H)$ loop is square, is linear. The larger the ratio *OD/ID* of the core, the lower the slope of Region 3.

Regions 4 and 5 (or Region 5 alone) describe the nonlinear portion of $\phi_d(F)$ from nonsaturation to saturation. These regions cover most of the static $\phi(F)$ curve of a ferrite core. In the case of a tape-wound core, the $\phi_d(F)$ "wing" that approaches saturation is likely to be described by Region 5 alone. Referring to Eq. (82) in Report 5, Region 4 is described by the function

$$\phi_d = \frac{(\phi_s + \phi_r)H_q}{(l_o - l_i)H_n} \left\{ \frac{F}{H_d^{min}} - l_i + F\left(\frac{1}{H_n} - \frac{1}{H_q}\right) \ell n \left[\frac{F\left(1 - \frac{H_n}{H_d^{min}}\right)}{F - H_n l_i} \right] \right\} - \phi_r \tag{265}$$

and Region 5 is described by the function

$$\phi_d = \frac{(\phi_s + \phi_r)H_q}{(l_o - l_i)H_n} \left[l_o - l_i + F\left(\frac{1}{H_n} - \frac{1}{H_q}\right) \ell n \left(\frac{F - H_n l_o}{F - H_n l_i} \right) \right] - \phi_r \quad , \tag{266}$$

in which H_n and H_q are material nonsaturation constants and

$$H_d^{min} = \frac{1}{4} \left[H_s - \sqrt{H_s^2 - 8\left(1 + \frac{\phi_r}{\phi_s}\right) H_a H_q} \right] \quad , \tag{267}$$

where

$$H_s = H_a + H_q + H_n + \frac{\phi_r}{\phi_s}(H_a + H_q - H_n) \quad . \tag{268}$$

Region 6 describes $\phi_d(F)$ in the positive saturation region. It is antisymmetric to Region 1 if the latter were extended to the region of negative F. Applying the antisymmetric relation $\phi_d(F) = -\phi_d(-F)$ to Eq. (264), Region 6 is then described by the function

$$\phi_{ds} = \frac{\phi_s - \phi_r}{(l_o - l_i)H_a} F \ln\left(\frac{F + H_a l_o}{F + H_a l_i}\right) + \phi_r \quad , \tag{269}$$

where $\phi_{ds}(F)$ denotes $\phi_d(F)$ in Region 6.

In order to decrease the computer-run time, we shall abbreviate the $\phi_d(F)$ expression of each region to its simplest form by lumping those terms that are constant throughout the run into a minimum number of newly defined *auxiliary parameters*. These auxiliary parameters will be computed once, in the beginning of the run, and then be used at each iteration. The auxiliary parameters are defined as follows for the general $\phi_d(F)$ model. As we shall see in Sec. C-3, some of these auxiliary parameters will be modified for certain simplified $\phi_d(F)$ models.

Regions 1 and 6:

$$p_1 = \frac{\phi_s - \phi_r}{(l_o - l_i)H_a} \tag{270}$$

$$p_2 = H_a l_o \tag{271}$$

$$p_3 = H_a l_i \tag{272}$$

$$p_4 = p_1(p_2 - p_3) \tag{273}$$

Region 2:

$$p_5 = p_1 \ln\left(\frac{F_{d1} - p_2}{F_{d1} - p_3}\right) F_{d1} - \phi_r \tag{274}$$

$$p_6 = p_1 \ln\left(\frac{F_{d1} - p_2}{F_{d1} - p_3}\right) + \frac{p_4 F_{d1}}{(F_{d1} - p_2)(F_{d1} - p_3)} \tag{275}$$

$$p_7 = \ell n \left[\frac{\phi_{d2} - p_5 - p_6(F_{d2} - F_{d1})}{0.5(\phi_{d2} - p_5) - p_6(F_{dz} - F_{d1})} \right] \Big/ \ell n \left(\frac{F_{d2} - F_{d1}}{F_{dz} - F_{d1}} \right) \quad (276)$$

$$p_8 = \frac{\phi_{d2} - p_5 - p_6(F_{d2} - F_{d1})}{(F_{d2} - F_{d1})^{p_7}} \quad (277)$$

$$p_9 = p_7 - 1.0 \quad (278)$$

Region 3:

$$p_{10} = \frac{\phi_{d3} - \phi_{d2}}{F_{d3} - F_{d2}} \quad (279)$$

$$p_{11} = p_{10} F_{d3} - \phi_{d3} \quad (280)$$

Regions 4 and 5:

$$p_0 = \frac{(\phi_s + \phi_r) H_q}{(l_o - l_i) H_n} \quad (281)$$

$$p_{12} = \frac{p_0}{H_d^{min}} \quad (282)$$

$$p_{13} = p_0 \left(\frac{1}{H_n} - \frac{1}{H_q} \right) \quad (283)$$

$$p_{14} = 1 - \frac{H_n}{H_d^{min}} \quad (284)$$

$$p_{15} = H_n l_i \quad (285)$$

$$p_{16} = \phi_r + p_0 l_i \quad (286)$$

$$p_{17} = H_n l_o \quad (287)$$

$$p_{18} = p_{16} - p_0 l_o \tag{288}$$

$$p_{19} = p_{13} p_{15} \tag{289}$$

$$p_{20} = p_{13}(p_{17} - p_{15}) \quad . \tag{290}$$

We now substitute Parameters p_1 through p_4 into Eq. (264), which describes Region 1, and Parameters p_0 and p_{12} through p_{20} into Eqs. (265) and (266), which describe Regions 4 and 5. In addition, we use Parameters p_5 through p_8 to formulate $\phi_d(F)$ of Region 2 and use Parameters p_{10} and p_{11} to formulate $\phi_d(F)$ of Region 3. The resulting abbreviated expressions for computing $\phi_d(F)$ and $\phi_d'(F) = d\phi_d(F)/dF$ at each iteration [see Eqs. (262) and (263)] are given as follows:

If $0 \leq F \leq F_{d1}$ (Region 1), then

$$q_1 = p_1 \ln\left(\frac{F - p_2}{F - p_3}\right) \quad , \tag{291}$$

$$\phi_d = q_1 F - \phi_r \quad , \tag{292}$$

and

$$\phi_d' = q_1 + \frac{p_4 F}{(F - p_2)(F - p_3)} \quad . \tag{293}$$

If $F_{d1} \leq F \leq F_{d2}$ (Region 2), then

$$q_2 = p_8 (F - F_{d1})^{p_9} \quad , \tag{294}$$

$$\phi_d = p_5 + (p_6 + q_2)(F - F_{d1}) \quad , \tag{295}$$

and

$$\phi_d' = p_6 + p_7 q_2 \quad . \tag{296}$$

If $F_{d2} \leq F \leq F_{d3}$ (Region 3), then

$$\phi_d = p_{10} F - p_{11} \tag{297}$$

and

$$\phi'_d = p_{10} \quad . \tag{298}$$

If $F_{d3} \leq F \leq F_{d4}$ (Region 4), then

$$q_4 = p_{13} \ln\left(\frac{p_{14}F}{F - p_{15}}\right) \quad , \tag{299}$$

$$\phi_d = (p_{12} + q_4)F - p_{16} \quad , \tag{300}$$

and

$$\phi'_d = p_{12} + q_4 - \frac{p_{19}}{F - P_{15}} \quad . \tag{301}$$

If $F_{d4} \leq F$ (Region 5), then

$$q_5 = p_{13} \ln\left(\frac{F - p_{17}}{F - p_{15}}\right) \quad , \tag{302}$$

$$\phi_d = q_5 F - p_{18} \quad , \tag{303}$$

and

$$\phi'_d = q_5 + \frac{p_{20}F}{(F - p_{15})(F - p_{17})} \quad . \tag{304}$$

Substitution of Parameters p_1 through p_3 into Eq. (269) results in the following abbreviated expressions for $\phi_d(F)$ and $\phi'_d(F)$ in Region 6:

$$q_6 = p_1 \ln\left(\frac{F + p_2}{F + p_3}\right) \tag{305}$$

$$\phi_{ds} = q_6 F + \phi_r \quad , \tag{306}$$

and

$$\phi'_{ds} = q_6 - \frac{p_4 F}{(F + p_2)(F + p_3)} \quad . \tag{307}$$

It is possible that $\phi > \phi_d$ during the switching time. However, unless ϕ is initially close to ϕ_r and the rise of $F(t)$ is extremely fast, ϕ cannot exceed ϕ_{ds} physically. But since computational errors may bring ϕ above ϕ_{ds}, we shall compute ϕ_{ds} if $\phi > \phi_d$, and if we find that $\phi > \phi_{ds}$, then we shall set $\phi = \phi_{ds}$. This procedure brings another point. The parameter values obtained by curve fitting of Region 5 may result in ϕ_d exceeding ϕ_{ds} in Region 6 at high values of F. Usually, this excess is negligible because both ϕ_d and ϕ_{ds} approach ϕ_s as $F \rightarrow \infty$. However, in order to be consistent with the requirement that $\phi \leq \phi_{ds}$, we shall set $\phi_{ds} = \phi_d$ if $\phi_{ds} < \phi_d$ on the basis of Eqs. (306) and (303).

Simplified Models - The general $\phi_d(F)$ model described above will quite often be simplified for specific cores, depending on the material, dimensions, and the temperature of these cores. Assuming that Regions 1 and 6 are nonlinear, six simplified models, referred to as Types A through F, are distinguished in Fig. 23(b). Note that no distinction is made between the case of both Regions 4 and 5 and the case of Region 5 alone because the two cases are related. Usually, polycrystalline ferrite cores are characterized by Types B and D, whereas grain-oriented tape-wound cores may exhibit any type. Let us examine each $\phi_d(F)$ type separately.

Type A - Region 2 is eliminated, and thus

$$F_{d2} = F_{d1}$$

and

$$\phi_{d2} = \phi_{d1} \quad .$$

Type B - Region 3 is eliminated, and thus

$$F_{d3} = F_{d2}$$

and

$$\phi_{d3} = \phi_{d2} \quad .$$

Type C - Regions 4 and 5 are eliminated because ϕ_{d3} at $F = F_{d3}$ is along Region 6, Eqs. (305) and (306), *i.e.*,

$$\phi_{d3} = \phi_{ds} \ @ \ F = F_{d3} \quad .$$

Type D - Regions 2 and 3 are eliminated, and thus

$$F_{d3} = F_{d2} = F_{d1} = H_d^{min} \ l_i$$

and

$$\phi_{d3} = \phi_{d2} = \phi_{d1} = \phi_{d1}^{min} \quad ,$$

where

$$\phi_{d1}^{min} = \phi_d \ @ \ F = H_d^{min} \ l_i \quad \text{in Region 1.}$$

Type E - Regions 2, 4, and 5 are eliminated, and thus

$$F_{d2} = F_{d1} \quad ,$$

$$\phi_{d2} = \phi_{d1} \quad ,$$

and

$$\phi_{d3} = \phi_{ds} \ @ \ F = F_{d3} \quad .$$

Type F - Regions 3, 4, and 5 are eliminated, and thus

$$F_{d3} = F_{d2}$$

and

$$\phi_{d3} = \phi_{d2} = \phi_{ds} \ @ \ F = F_{d2} \quad .$$

ii. $\dot{\phi}_p(F)$ Model

A general model for the $\dot{\phi}_p(F)$ curve is shown in Fig. 24 and is expressed as follows:

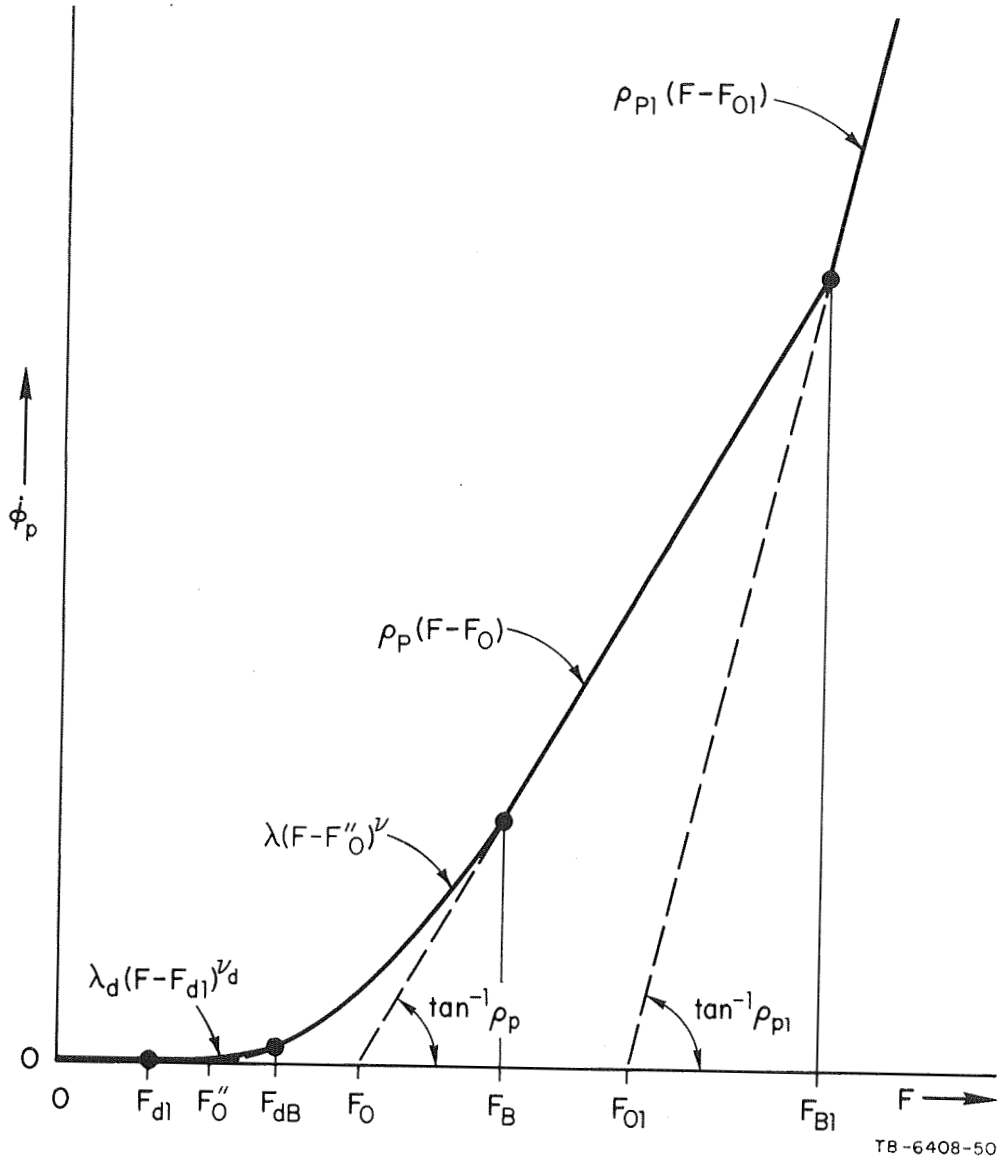


Figure 24 A GENERAL MODEL FOR THE $\dot{\phi}_p(F)$ CURVE

$$\dot{\phi}_p = \begin{cases} 0 & \text{for} \quad 0 \le F \le F_{d1} \quad (308a) \\ \lambda_d (F - F_{d1})^{\nu_d} & \text{for} \quad F_{d1} \le F \le F_{dB} \quad (308b) \\ \lambda (F - F_0'')^{\nu} & \text{for} \quad F_{dB} \le F \le F_B \quad (308c) \\ \rho_p (F - F_0) & \text{for} \quad F_B \le F \le F_{B1} \quad (308d) \\ \rho_{p1} (F - F_{01}) & \text{for} \quad F_{B1} \le F \quad . \quad (308e) \end{cases}$$

This model is identical with the one used in Report 5, Eq. (81), except for the addition of another linear $\dot{\phi}_p(F)$ region for $F \ge F_{B1}$. This region has been observed with some tape-wound metallic cores. For most cases, however, this additional region is unnecessary and will be avoided automatically by assigning an arbitrary high value to F_{B1}, e.g., $F_{B1} = 10^{30}$.

Expressions for $\dot{\phi}_p'$, to be used in Eq. (263), are obtained by differentiating Eqs. (308) with respect to F. We, thus, obtain

$$\dot{\phi}_p' = \begin{cases} 0 & \text{for} \quad 0 \le F \le F_{d1} \quad (309a) \\ \lambda_d (F - F_{d1})^{\nu_d} \nu_d / (F - F_{d1}) & \text{for} \quad F_{d1} \le F \le F_{dB} \quad (309b) \\ \lambda (F - F_0'')^{\nu} \nu / (F - F_0'') & \text{for} \quad F_{dB} \le F \le F_B \quad (309c) \\ \rho_p & \text{for} \quad F_B \le F \le F_{B1} \quad (309d) \\ \rho_{p1} & \text{for} \quad F_{B1} \le F \quad . \quad (309e) \end{cases}$$

Nine parameters (λ_d, ν_d, F_{dB}, F_0'', λ, ν, F_B, F_0, and ρ_p) define Regions 1 through 4 of the $\dot{\phi}_p(F)$ curve. We shall now show that the values of some of these parameters may be approximated quite closely from the values of the remaining parameters.

For continuity, the expressions for $\dot{\phi}_p(F)$ and $d\dot{\phi}_p(F)/dF$ of neighboring regions must be equal at the borders $F = F_{dB}$ and $F = F_B$. Continuity at $F = F_{dB}$ imposes the relations

$$\nu_d = \nu \frac{F_{dB} - F_{d1}}{F_{dB} - F_0''} \quad (310)$$

and

$$\lambda_d = \lambda \frac{(\nu/\nu_d)^{\nu}}{(F_{dB} - F_{d1})^{\nu_d - \nu}} \quad ; \tag{311}$$

continuity at $F = F_B$ imposes the relations

$$\nu = \frac{F_B - F_0''}{F_B - F_0} \tag{312}$$

and

$$\lambda = \frac{\rho_p}{\nu(F_B - F_0'')^{\nu-1}} \quad . \tag{313}$$

These relations impose four constraints on the nine parameters in Eqs. (308). Hence, only five parameters are needed to completely specify $\dot{\phi}_p$ *vs.* F.

Further reduction in the number of parameters that need to be specified is based on the empirical relation

$$F_{dB} \approx 1.15\, F_0'' \quad . \tag{314}$$

If the values of F_0'', F_B, F_0, and ρ_p are known, Eqs. (312) and (313) may be used to compute ν and λ. Knowing the values of F_{d1}, F_0'', λ, and ν, we use Eqs. (314), (310), and (311) to compute F_{dB}, ν_d, and λ_d in that order.

c. Summary

Let us summarize the computation involved in evaluation of $\dot{\phi} = \dot{\phi}_\epsilon + \dot{\phi}_i$, Eq. (227), and $\dot{\phi}' = \dot{\phi}'_\epsilon + \dot{\phi}'_i$, Eq. (228), on the basis of the square loop core model.

During the first three time steps, we compute at each iteration $\dot{\phi}_\epsilon \approx \epsilon(F)[F - F_{(-1)}]/\Delta t$, Eq. (234), and $\dot{\phi}'_\epsilon \approx \{\epsilon + \epsilon'[F - F_{(-1)}]\}/\Delta t$, Eq. (235), where $\epsilon(F)$ and $\epsilon'(F)$ are computed from Eqs. (230) and (236). During the following time steps, we compute $\dot{\phi}_\epsilon \approx \epsilon(F)\dot{\bar{F}}$, Eq. (253), and

$\dot{\phi}'_\epsilon = \epsilon'(F)\bar{\dot{F}} - 3\epsilon(F)/S_t$, Eq. (254), where $\bar{\dot{F}}$ is the average slope of the previous three $F(t)$ points, Eqs. (245) through (251), and S_t is computed from Eq. (245). The expressions for $\epsilon'(F)$ include three auxiliary parameters, p_1, p_2, and p_3, which depend on the static core parameters according to Eqs. (231) through (233).

Computation of $\dot{\phi}_i$ is based on Eq. (262), which includes ϕ_d and $\dot{\phi}_p$; computation of $\dot{\phi}'_i$ is based on Eq. (263), which includes ϕ_d, ϕ'_d, $\dot{\phi}_p$, and $\dot{\phi}'_p$. Expressions for ϕ_d and ϕ'_d as functions of F are given in Eqs. (291) through (306) for a general $\phi_d(F)$ model. These expressions include twenty auxiliary parameters, p_1 through p_{20}, which depend on the static core parameters, according to Eqs. (270) through (290). In practice, six simplified $\phi_d(F)$ types, Types A through F, are obtained by eliminating one to three regions from the general $\phi_d(F)$ model (Type G). Expressions for $\dot{\phi}_p$ and $\dot{\phi}'_p$ as functions of F are given in Eqs. (308) and (309), respectively, for a general, five-region $\dot{\phi}_p(F)$ model. These expressions include the static parameter F_{d1} and twelve dynamic parameters: λ_d, ν_d, F_{dB}, λ, F''_0, ν, F_B, ρ_p, F_0, F_{B1}, ρ_{p1}, and F_{01}. In practice, $\dot{\phi}_p(F)$ may be simplified by eliminating some of these regions, thus decreasing the number of dynamic parameters. The values of ν, λ, F_{dB}, ν_d, and λ_d may be computed quite accurately from the values of F''_0, F_B, F_0, and ρ_p.

3. Input Core Data

The input core data are used to determine the static and dynamic core parameters for normal and worst-case conditions. The general $\phi_d(F)$ parameters include F_{d1}, F_{d2}, F_{d3}, F_{d4}, and the auxiliary parameters p_1 through p_{20}. Depending on the $\phi_d(F)$ type, some of these parameters may be superfluous. The general $\dot{\phi}_p(F)$ parameters are λ_d, ν_d, F_{dB}, λ, F''_0, ν, F_B, ρ_p, F_0, F_{B1}, ρ_{p1}, and F_{01}. Some of these parameters may also be superfluous, depending on the shape of the $\dot{\phi}_p(F)$ curve.

Two kinds of input core data have been distinguished so far:

(1) Data based on measured $\phi_d(F)$ and $\dot{\phi}_p(F)$ curves. Such data have been used in Reports 1 through 5 for ferrite cores, and will be designated by "ϕF".

(2) Data based on static and dynamic $B(H)$ loops. Such data are found in a manufacturer's catalog of tape-wound cores, and will be designated by "BH",

Let us discuss the data entry and the preliminary computation associated with each type separately.

a. $\phi_d(F)$ and $\dot{\phi}_p(F)$ Data (ϕF data)

i. Input Data

Except for core dimensions, the input core data are entered in MKS units.

Core dimensions

OD (Outside Diameter), in inch.

ID (Inside Diameter), in inch.

$\phi_d(F)$ data (see Fig. 23)

For the general $\phi_d(F)$ model, Fig. 23(a), the following parameters are specified:

ϕ_s; ϕ_r;

Type of $\phi_d(F)$ curve (A, B, C, D, E, F, or G);

Core material (Ferrite or Tape-wound);

F_{d1}; F_{dz}; F_{d2}; F_{d3};

ϕ_{d2};

H_a; H_q; H_n .

If Region 2 is linear (with finite or infinite slope), then F_{dz} need not be specified. If Region 2 is linear and has an infinite slope (jump), then F_{d2} need not be specified.

For simplified $\phi_d(F)$ models, Fig. 23(b), the parameters belonging to the nonexisting regions are not specified, as shown in Table I. The value of ϕ_r must be specified for any type.

Table I

STATIC $\phi(F)$ PARAMETERS TO BE SPECIFIED FOR EACH $\phi_d(F)$-MODEL TYPE

(See Fig. 23). ✓ - specified; (✓) - specified if Region 2 is nonlinear (need not, but may be specified otherwise); -- - not specified.

PARAMETER \ $\phi_d(F)$-MODEL Type	G (General)	A	B	C	D	E	F
F_{d1}	✓	✓	✓	✓	--	✓	✓
F_{dz}	(✓)	--	(✓)	(✓)	--	--	(✓)
F_{d2}	(✓)	--	(✓)	(✓)	--	--	(✓)
F_{d3}	✓	✓	--	✓	--	✓	--
ϕ_{d2}	✓	--	✓	✓	--	--	--
H_a	✓	✓	✓	✓	✓	✓	✓
H_q	✓	✓	✓	--	✓	--	--
H_n	✓	✓	✓	--	✓	--	--

$\dot{\phi}_p(F)$ Parameters (see Fig. 24)

For the general $\dot{\phi}_p(F)$ model, Fig. 24, the following parameters are specified:

$$\lambda_d;\ \nu_d;\ F_{dB};$$

$$F_0'';\ \lambda;\ \nu;\ F_B;$$

$$F_0;\ \rho_p;\ F_{B1};$$

$$F_{01};\ \rho_{p1}\ .$$

If Region 2 ($F_{d1} \leq F \leq F_{dB}$) is unknown, then λ_d, ν_d, and F_{dB} are not specified. If Region 5 ($F_{B1} \leq F$) is absent, then F_{B1}, F_{01}, and ρ_{p1} are not specified.

ii. Preliminary Computation

The steps for determining the required parameters are outlined as follows:

$\phi_d(F)$ Parameters

(1) Polycrystalline ferrite cores are characterized by $\phi_d(F)$ curves of type B or D. Anticipating more

use with ferrite cores, and since the former is more general, we shall arbitrarily assume that if no type is specified, then Type B is implied.

(2) If $\phi_s = 0$ (*i.e.*, not specified), then $\phi_s = 1.1\ \phi_r$.

(3) If a d_{max}-percent variation of the input-parameter values for a worst-case analysis is specified by the user, then the sign S of the change in each input parameter must also be provided. The S-data are entered following a normal data entry. Each parameter of nominal value p_n is then changed according to

$$p = p_n \left(1 + S \frac{d_{max}}{100}\right) \quad . \tag{315}$$

(4) The lengths $l_o = \pi \cdot 0.0254 \cdot \mathrm{OD}$ and $l_i = \pi \cdot 0.0254 \cdot \mathrm{ID}$ are computed.

(5) Auxiliary parameters p_1 through p_4 are computed from Eqs. (270) through (273).

(6) If Type = F, then

If $F_{d2} = 0$ (*i.e.*, when only F_{d1} is specified), then $F_{d2} = F_{d1}$. Substituting $F = F_{d2}$ into Eqs. (305) and (306), we compute

$$\phi_{d2} = p_1 \ln\left(\frac{F_{d2} + p_2}{F_{d2} + p_3}\right) F_{d2} + \phi_r \quad . \tag{316}$$

(7) If Type = G, B, C, or F (*i.e.*, Region 2 exists), then

If $F_{dz} = 0$ *or* $F_{dz} = F_{d1}$ *or* $F_{dz} = 0.5\ (F_{d1} + F_{d2})$ (*i.e.*, Region 2 is linear), then

If $F_{d2} = 0$ *or* $F_{d2} = F_{d1}$ (*i.e.*, Region 2 is a jump), then

$$F_{d2} = F_{d1}.$$

In order to prevent computational oscillations, a jump is converted into a straight line of high slope by setting

$$F_{d1} = 0.95\ F_{d2} \tag{317}$$

$$F_{dz} = 0.5\ (F_{d1} + F_{d2}) \tag{318}$$

p_5 is computed from Eq. (274).

Reducing Eq. (295) to $\phi_d = p_5 + p_6(F - F_{d1})$ and substituting $\phi_d = \phi_{d2}$ @ $F = F_{d2}$,

$$p_6 = \frac{\phi_{d2} - p_5}{F_{d2} - F_{d1}} \tag{319}$$

To assure that $q_2 = 0$, set $p_7 = 1.0$ and $p_8 = p_9 = 0$.

Else (*i.e.*, Region 2 is nonlinear), then p_5 through p_9 are computed from Eqs. (274) through (278).

(8) If Type = *G*, *A*, *B*, or *D* (*i.e.*, Region 4 and/or 5 exist), then H_s and H_d^{min} are computed from Eqs. (268) and (267), and p_0 and p_{12} through p_{20} are computed from Eqs. (281) through (290). Also, if Type = *D*, then

$$F_{d1} = H_d^{min}\, l_i$$

$$F_{d4} = H_d^{min}\, l_o \quad . \tag{320}$$

Else (*i.e.*, Type = *C*, *E*, or *F*), then Region 4 is bypassed (because $F_{d4} = 0$), and Region 5 is replaced by Region 6 by artificially setting $p_{13} = p_1$, $p_{15} = -p_3$, $p_{17} = -p_2$, $p_{18} = -\phi_r$, and $p_{20} = -p_4$ [*cf.* Eqs. (302) through (307)].

(9) If Type = *G*, *A*, *C*, or *E* (*i.e.*, Region 3 exists), then

If Type = *G* or *A* (*i.e.*, Regions 4 and/or 5 exist), then

If $F_{d3} < F_{d4}$ (*i.e.*, Region 4 exists), then ϕ_{d3} is computed by substituting $F = F_{d3}$ into $\phi_d(F)$ in Region 4, Eqs. (299) and (300).

Else (*i.e.*, Region 4 does not exist), then ϕ_{d3} is computed by substituting $F = F_{d3}$ into $\phi_d(F)$ in Region 5, Eqs. (302) and (303).

Else (*i.e.*, Type = *C* or *E*), then ϕ_{d3} is computed by substituting $F = F_{d3}$ into $\phi_d(F)$ in Region 6, Eqs. (305) and (306).

If Type = *G* or *C* (*i.e.*, Region 2 exists), then ϕ_{d2} is known, and the slope of Region 3 is

$$p_{10} = (\phi_{d3} - \phi_{d2})/(F_{d3} - F_{d2}) \tag{321}$$

Else (*i.e.*, Region 2 does not exist), then $p_{10} = (\phi_{d3} - \phi_{d1s})/(F_{d3} - F_{d1})$, where ϕ_{d1s} is computed by substituting $F = F_{d1}$ into $\phi_d(F)$ in Region 1, Eqs. (291) and (292).

Using Eq. (297), $p_{11} = p_{10}F_{d3} - \phi_{d3}$.

$\dot{\phi}_p(F)$ Parameters

(1) If $F_{B1} = 0$ (not specified), then Region 5 of $\dot{\phi}_p(F)$ is assumed not to exist, and F_{B1} is set to 10^{30} (in order to prevent entering Region 5 if $F > F_B$).

(2) If $F_0'' < F_{d1}$, we assume that F_0'' is erroneous and set $F_0'' = F_{d1}$.

(3) If $\nu = 0$ (not specified), then, following Eq. (312),

$$\nu = \frac{F_B - F_0''}{F_B - F_0} .$$

(4) If $\lambda = 0$ (not specified), then, following Eq. (313),

$$\lambda = \frac{\rho_p}{\nu(F_B - F_0'')^{\nu-1}} .$$

(5) If $F_{dB} = 0$ (not specified), then, following Eq. (314), $F_{dB} = 1.15\ F_0''$.

(6) If $\nu_d = 0$ (not specified), then, following Eq. (310),

$$\nu_d = \nu \frac{F_{dB} - F_{d1}}{F_{dB} - F_0''} .$$

(7) If $\lambda_d = 0$ (not specified), then, following Eq. (311),

$$\lambda_d = \lambda \frac{(\nu/\nu_d)^{\nu}}{(F_{dB} - F_{d1})^{\nu_d - \nu}} .$$

(8) A "negligible value" of $\dot{\phi}$ is estimated on the basis of Eq. (257) to be

$$ZV = 0.003\lambda(0.3 \cdot F_0'')^{\nu} . \tag{322}$$

b. Static and Dynamic $B(H)$-Loop Data (BH Data)

In the absence of measured $\phi_d(F)$ and $\dot{\phi}_p(F)$ parameters, it is possible to estimate the values of these parameters on the basis of static and dynamic $B(H)$ loops. Such data are usually available for tape-wound cores in the manufacturer's catalog. An example is shown in Fig. 25 (the shapes of the curves correspond to Square Permalloy 80 material[19]). Quite commonly, the catalog data are given in the following units: H in oersted, B in gausses, effective cross-sectional area of the core in cm^2, and the core diameters in inch. The input data will, therefore, use such units, and be designated by subscript "in", before being converted into MKS units. The resulting parameter values will, of course, be less accurate than measured ones.

i. Input Data

Core Dimensions

OD (Outside Diameter), in inch

ID (Inside Diameter), in inch

A_{in} (effective cross-sectional core area), in cm^2

Static $B(H)$ Data [See Fig. 25(a)]

For the general $B_d(H)$ model, Fig. 25(a), the following data are specified in emu units of gausses for B and in oersted for H:

$B_{r,in}$;

Type of $B_d(H)$ curve (A, B, C, D, E, F, or G);

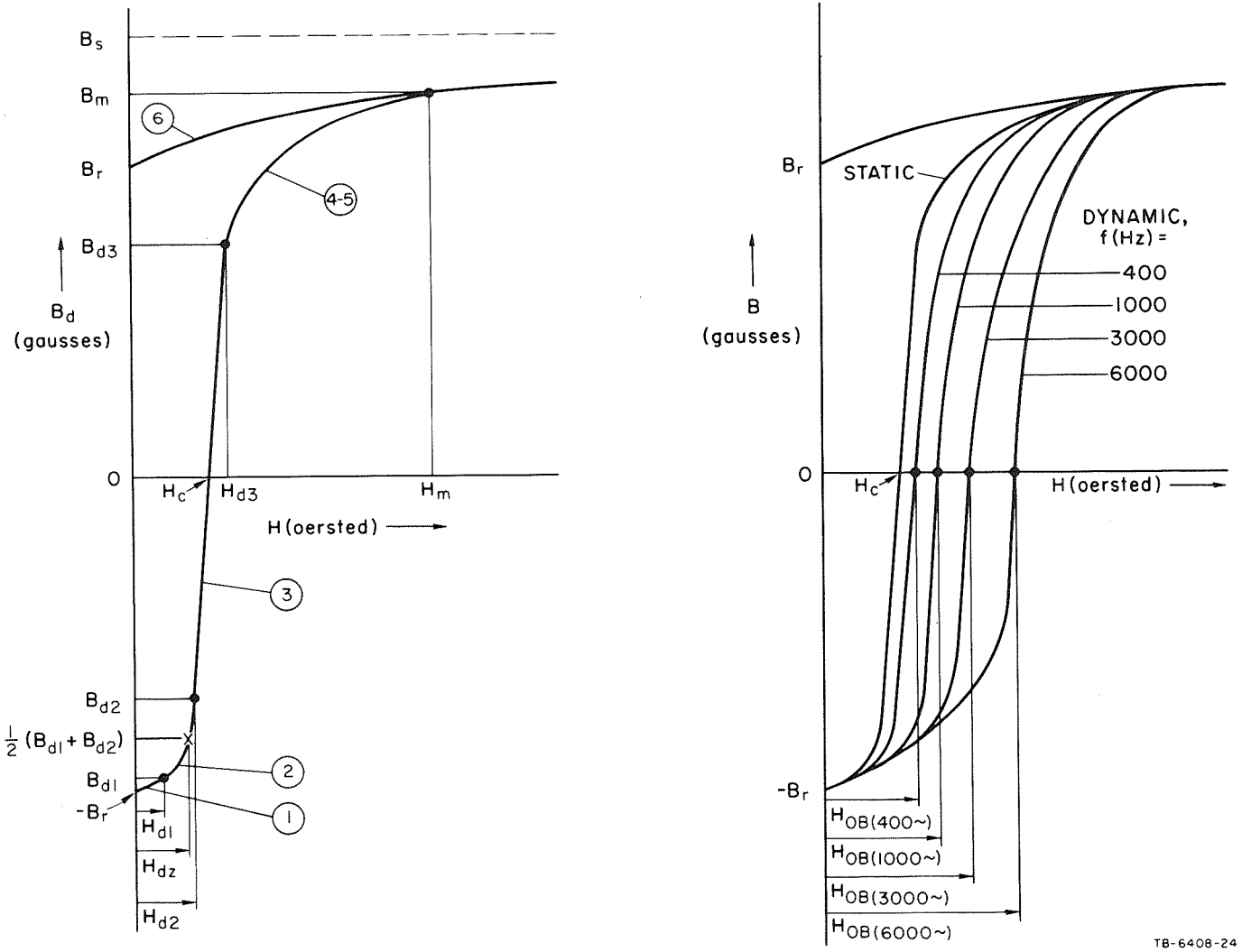
Core material (Ferrite or Tape-wound);

$H_{d1,in}$; $H_{dz,in}$; $H_{d2,in}$; $H_{d3,in}$; $H_{m,in}$;

$B_{d2,in}$; $B_{d3,in}$; $B_{m,in}$.

Simplified $B_d(H)$ models are classified into Types A through F, which are similar in shape to the $\phi_d(F)$-model types in Fig. 23(b). These $B_d(H)$-model types are, therefore, defined in Fig. 23(b) if ϕ is replaced by B and F is replaced by H.

If Region 2 is linear (with finite or infinite slope), then $H_{dz,in}$ need not be specified. If Region 2 is linear and has an infinite slope (jump), then $H_{d2,in}$ need not be specified.



(a) STATIC B(H) CURVE

(b) STATIC AND DYNAMIC B(H) CURVES

Figure 25 STATIC AND DYNAMIC B(H) CURVES OF A TAPE-WOUND-CORE MATERIAL

For simplified $B_d(H)$-model types, the parameters belonging to nonexisting regions are not specified, as shown in Table II. For Type *D*, although the values of $B_{d3,in}$ and $H_{d3,in}$ correspond to any arbitrary point on the curve, it is preferred that $B_{d,in} = 0$ and, consequently, $H_{d3,in} = H_{c,in}$. The value of $B_{r,in}$ must be specified for any $B_d(H)$-model type.

Table II

STATIC $B(H)$ PARAMETERS TO BE SPECIFIED FOR EACH $B_d(H)$-MODEL TYPE [see Figs. 23 and 25(a)]. √ - specified; (√) - specified if Region 2 is nonlinear (need not, but may be specified otherwise); -- - not specified.

PARAMETER \ $B_d(H)$-MODEL TYPE	*G* (General)	*A*	*B*	*C*	*D*	*E*	*F*
$H_{d1,in}$	√	√	√	√	--	√	√
$H_{dz,in}$	(√)	--	(√)	(√)	--	--	(√)
$H_{d2,in}$	(√)	--	(√)	(√)	--	--	(√)
$H_{d3,in}$	√	√	--	√	√	√	--
$H_{m,in}$	√	√	√	--	√	--	--
$B_{d2,in}$	√	--	√	√	--	--	--
$B_{d3,in}$	√	√	--	--	√	--	--
$B_{m,in}$	√	√	√	--	√	--	--

Dynamic $B(H)$ Data [see Fig. 25(b)]

n_{dynp} [number of dynamic $B(H)$ curves]

For $j = 1, 2, \ldots, n_{dynp}$:

$f_{(j)}$ [frequency (in Hz)]; $H_{0B,in(j)}$ [H (in oersted) @ $B = 0$].

ii. Computation of Approximate $\phi_d(F)$ and $\dot{\phi}_p(F)$ Parameters

Before any computation is performed, the input data are modified according to Eq. (315), if a worst-case parameter variation of d_{max} percent is specified by the user. We shall next show how the above data, which are extracted from static and dynamic $B(H)$ loops, may be utilized to compute approximate values for the $\phi_d(F)$ and $\dot{\phi}_p(F)$ input parameters, Sec. I-C-a-i, pp. 64-65.

$\phi_d(F)$ Parameters

The lengths $l_o = \pi \cdot 0.0254 \cdot \mathrm{OD}$ and $l_i = \pi \cdot 0.0254 \cdot \mathrm{ID}$ are first computed.

The input static $B(H)$ data are then converted to MKS units, using the relations

$$A(m^2) = A_{in}(cm^2) \cdot 10^{-4} \quad , \tag{323}$$

$$B(\mathrm{weber}/m^2) = B_{in}(\mathrm{gausses}) \cdot 10^{-4} \quad , \tag{324}$$

and

$$H(At/m) = H_{in}\ (\mathrm{oersted}) \cdot 79.577 \quad . \tag{325}$$

For Types G (general), A, B, and D, we assume that $B_m \approx (B_s + B_r)/2$, and hence,

$$B_s = 2B_m - B_r \quad . \tag{326}$$

For Types C, E, and F, we simply let $B_s = 1.1\ B_r$.

Thus,

$$\phi_s = B_s A \quad , \tag{327}$$

$$\phi_r = B_r A \quad , \tag{328}$$

and

$$\phi_{d2} = B_{d2} A \quad . \tag{329}$$

Comparing Figs. 23 and 25(a), we assume that, except for Type F, ϕ_{d2} is close to $-\phi_r$ and that ϕ_{d3} is close to ϕ_r; hence,

$$F_{d1} = H_{d1} l_i \tag{330}$$

$$F_{dz} \approx H_{dz} l_i \tag{331}$$

$$F_{d2} \approx H_{d2} l_i \tag{332}$$

and

$$F_{d3} \approx H_{d3} l_o \quad . \tag{333}$$

In the case of Type F, $H_{d3} = H_{d2}$, and hence,

$$F_{dz} \approx 0.5\, H_{dz}(l_o + l_i) \tag{331a}$$

and

$$F_{d2} \approx H_{d2} l_o \quad . \tag{332a}$$

Let us use the hyperbolic models for the static $B(H)$ curve in Report 2, Eqs. (10) and (11), to describe $B_d(H)$ in Regions 1 and 4-5. Accordingly, Region 1 is described by

$$B_d = -B_r - (B_s - B_r)\frac{H}{H - H_a} \tag{334}$$

and, since $(B_s + B_r) \approx 2B_m$, Region 4-5 is described by

$$B_d = -B_r + 2B_m \frac{H - H_q}{H - H_n} \quad . \tag{335}$$

Using the relation $B_d(H) = -B_d(-H)$ in the saturation regions, Region 6 is, then, described by

$$B_d = B_r + (B_s - B_r)\frac{H}{H + H_a} \quad . \tag{336}$$

Substituting $H = H_m$ and $B_d = B_m = (B_r + B_s)/2$ into Eq. (336), we find that for Types G, A, B, and D,

$$H_a = H_m \quad . \tag{337}$$

On the basis of some observation, we assume that for Types C, E, and F,

$$H_a = 4H_{d3} \quad . \tag{338}$$

Consider again Types G, A, B, and D [*i.e.*, static $B(H)$ curves with the nonlinear Region 4-5], and let $H_{d3} = H_{d2}$ and $B_{d3} = B_{d2}$ for Type B. Thus, for each of these four types, the coordinates (H_m, B_m) and (H_{d3}, B_{d3}) of two points in Region 4-5 are known. Substituting these coordinates into Eq. (335), we obtain two equations from which H_n and H_q are solved for. We thus obtain

$$H_n = \frac{H_{d3}(B_s - B_{d3}) - H_m(B_m - B_r)}{B_m - B_{d3}} \tag{339}$$

and

$$H_q = H_{d3} - \frac{B_{d3} + B_r}{2B_m}(H_{d3} - H_n) \quad . \tag{340}$$

This completes the computation of approximate values for the $\phi_d(F)$-model parameters ϕ_s, ϕ_r, F_{d1}, F_{dz}, F_{d2}, F_{d3}, ϕ_{d2}, $H_{\dot{a}}$, $H_{\dot{q}}$, and H_n. The preliminary computation of the auxiliary parameters p_1 through p_{20} now follows steps (5) through (9), Sec. I-C-3-a-ii, pp. 65-68.

$\dot{\phi}_p(F)$ Parameters

The dynamic $B(H)$ curves in Fig. 25(b) are commonly obtained while the tested core is excited by a sinusoidal voltage source e_s of amplitude E_{sp} and frequency f across N_d turns. Thus, assuming a uniform B across the core,

$$e_s = E_{sp} \sin(\omega t) = N_d A \dot{B}(t) \tag{341}$$

where

$$\omega = 2\pi f = \pi/T \quad ; \tag{342}$$

hence,

$$\dot{B}(t) = \dot{B}_p \sin(\omega t) \quad , \tag{343}$$

where

$$\dot{B}_p = \frac{E_{sp}}{N_d A} \quad . \tag{344}$$

Referring to Fig. 25, let us assume that E_{sp} and N_d are adjusted so that B swings between $-B_m$ and $+B_m$; hence, integration of Eq. (343) gives

$$B(t) = -B_m + \frac{\dot{B}_p}{\omega}[1 - \cos(\omega t)] \quad . \tag{345}$$

At $t = T$, $\cos(\omega t) = -1$ and $B(t) = B_m$; hence,

$$\dot{B}_p = \omega B_m = 2\pi f B_m \quad . \tag{346}$$

The waveforms of $\dot{B}(t)$ and $B(t)$ and the corresponding magnetizing field $H(t)$ are shown in Fig. 26. As is evident from the dynamic $B(H)$ curves in Fig. 25(b), the higher the frequency of excitation f, the faster the switching rate and, hence, the higher the required excess field $H - H_0$.

Note in Fig. 26 that at $t = T/2$, $\dot{B} = \dot{B}_p$, $B = 0$, and $H = H_{0B}$. Values of H_{0B} for different values of f can be obtained directly from the plots of dynamic $B(H)$ curves, Fig. 25(b). The corresponding $\dot{B}_p$ values can be computed from Eq. (346). Recalling that there are n_{dynp} dynamic $B(H)$ loops, we thus obtain n_{dynp} points for $\dot{B}_p$ *vs.* H, as shown in Fig. 27(a). We assume that these points correspond to the linear portion of the $\dot{B}_p(H)$ curve,

$$\dot{B}_p = \zeta_p (H - H_0) \quad , \tag{347}$$

where $\zeta_p = \rho_p l/A$ and $H_0 = F_0/l$ [see Report 2, p. 39, Eqs. (88) and (89)]. This allows us to determine ζ_p and H_0 by least-mean-square-error fitting of Eq. (347) and the n_{dynp} points in Fig. 27(a). We thus obtain

$$\zeta_p = 2\pi B_m \frac{n_{dynp} \sum_j f_{(j)} H_{0B(j)} - \sum_j f_{(j)} \sum_j H_{0B(j)}}{n_{dynp} \sum_j H_{0B(j)}^2 - \left[\sum_j H_{0B(j)}\right]^2} \tag{348}$$

and

$$H_0 = \frac{1}{n_{dynp}} \left[\sum_j H_{0B(j)} - \frac{2\pi B_m}{\zeta_p} \sum_j f_{(j)} \right] \quad , \tag{349}$$

where $\sum_j \equiv \sum_{j=1}^{n_{dynp}}$. Substituting

$$z_1 = \sum_j f_{(j)} \quad , \tag{350}$$

$$z_2 = \sum_j H_{0B(j)} \quad , \tag{351}$$

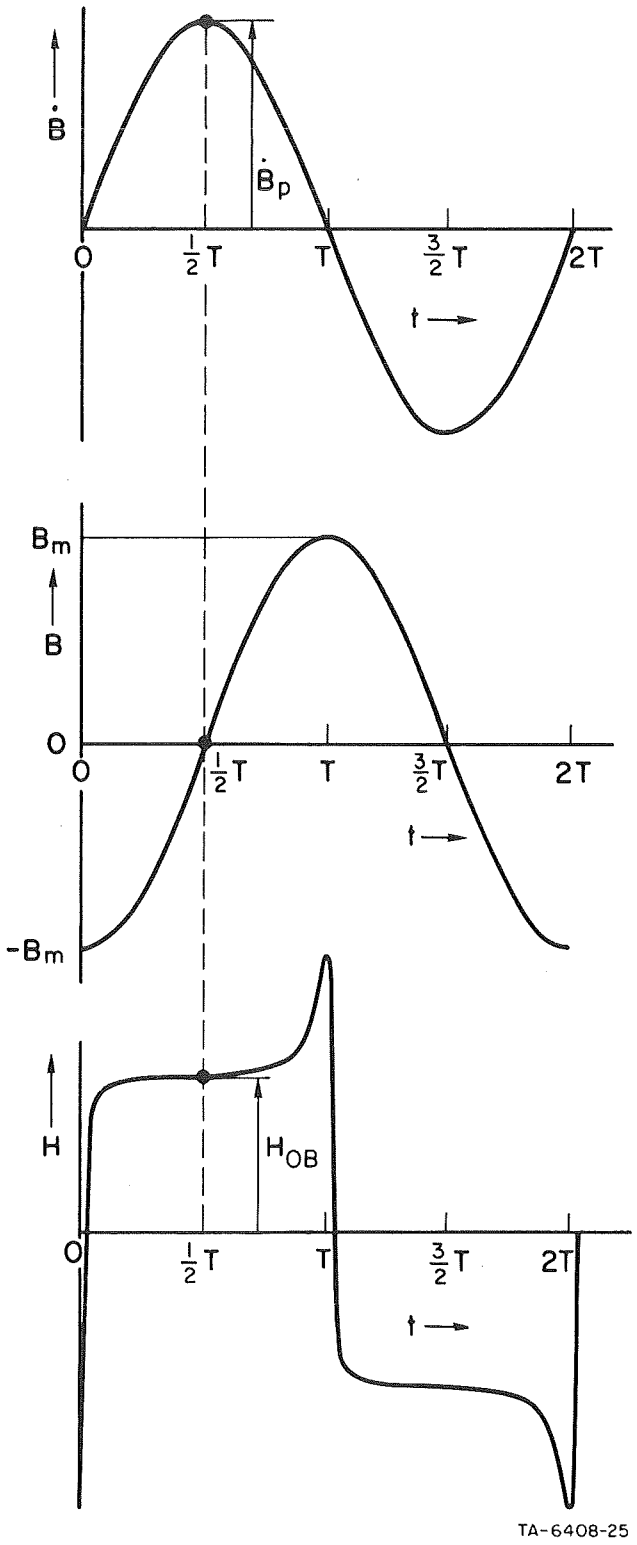
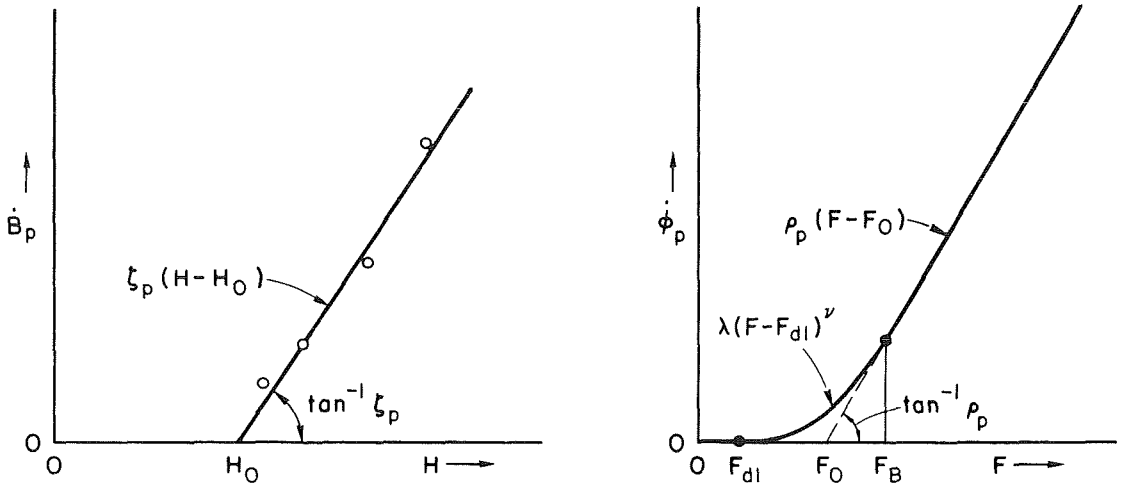


Figure 26 SINUSOIDAL $\dot{B}(t)$ AND THE RESULTING B(t) AND H(t) WAVEFORMS



(a) Fitting of $\dot{B}_p = \zeta_p(H - H_0)$ to $\dot{B}_p(H)$ points ($\dot{B}_p = 2\pi B_m f$; $H = H_{OB}$) obtained from a manufacturer's catalog

(b) A three-region $\dot{\phi}_p(F)$ curve in which $\rho_p = \zeta_p A/l$, $F_0 = H_0 l$, $F_B = F_0(a + bT)$; $\nu = (F_B - F_{d1})/(F_B - F_0)$, and $\lambda = \rho_p/[\nu(F_B - F_{d1})^{\nu} - 1]$

TA-6408-26

Figure 27 DETERMINATION OF APPROXIMATE $\dot{\phi}_p(F)$ PARAMETERS FROM DYNAMIC B(H) LOOPS

$$z_3 = \sum_j f_{(j)} H_{0B(j)} \quad , \tag{352}$$

$$z_4 = \sum_j H_{0B(j)}^2 \quad , \tag{353}$$

and

$$z_5 = \frac{n_{dynp} z_3 - z_1 z_2}{n_{dynp} z_4 - z_2^2} \tag{354}$$

into Eqs. (348) and (349), we get

$$\rho_p = 2\pi B_m z_5 \frac{A}{l} \tag{355}$$

and

$$F_0 = \frac{l}{n_{dynp}} \left(z_2 - \frac{z_1}{z_5} \right) \quad , \tag{356}$$

where

$$l = (l_o + l_i)/2 \quad . \tag{357}$$

An approximate value for F_B may be obtained from the empirical relations

$$F_B = \begin{cases} F_0[1.33 - 0.0007\ (T°K - 298)] & \text{for tape-wound cores} \quad (358a) \\ F_0[2.00 + 0.008\ (T°K - 298)] & \text{for ferrite cores.} \quad (358b) \end{cases}$$

These relations are based on averaging values of F_B/F_0 *vs.* temperature of tape-wound and ferrite cores whose parameters have been measured in the past. Note that F_B/F_0 decreases with temperature for tape-wound cores, whereas for ferrite cores, F_B/F_0 increases with temperature. These relations may be modified in future as more data are collected.

The general $\dot{\phi}_p(F)$ model, Eqs. (308), is simplified into a three-region curve

$$\dot{\phi}_p = \begin{cases} 0 & \text{for} \quad 0 \le F \le F_{d1} \quad (359a) \\ \lambda\ (F - F_{d1})^{\nu} & \text{for} \quad F_{d1} \le F \le F_B \quad (359b) \\ \rho_p\ (F - F_0) & \text{for} \quad F_B \le F\ , \quad (359c) \end{cases}$$

which is shown in Fig. 27(b). We now require that $\dot{\phi}_p(F)$ be continuous, *i.e.*, that the neighboring regions at $F = F_B$ have the same value of $\dot{\phi}_p$ and the same value of $d\dot{\phi}_p/dF$. Hence,

$$\nu = \frac{F_B - F_{d1}}{F_B - F_0} \quad (360)$$

and

$$\lambda = \frac{\rho_p}{\nu(F_B - F_{d1})^{\nu-1}} \quad . \quad (361)$$

Since F_{dB} is not assigned value, its value is zero by default. Therefore, the general $\dot{\phi}_p(F)$ model, Eqs. (308), becomes equivalent to the simplified $\dot{\phi}_p(F)$ model, Eqs. (359), by setting $F_0'' = F_{d1}$ and $F_{B1} = 10^{30}$. Since Region 2 of the general model is bypassed and Region 5 is never reached, the values of λ_d, ν_d, F_{01}, and ρ_{p1} need not be computed (they stay zero by default).

A "negligible value," ZV, is computed from Eq. (257).

II COMPUTER PROGRAM

The original Autonetics' TRAC program (which was received at Stanford Research Institute in August 1967) could not handle square-loop magnetic cores properly. In order to be able to analyze digital magnetic circuits, a core subprogram has been developed on the basis of the core model, Sec. I-C, pp. 40-78, and incorporated into the TRAC program. In addition, the program has been modified by including the provisions for the following tasks:

(1) Divide the transient response into modes of operation.

(2) Save the values of magnetic fluxes, inductor currents, or all the time variables from one mode to another.

(3) Make accessible to the user new auxiliary variables, such as F, ϕ, and ϕ of each core and the current through any circuit element.

(4) Plot any variable, using units, scales, and frame sizes which are either specified by the user (*e.g.*, for comparing computed and experimental waveforms), or determined automatically by the program on the basis of extreme values.

(5) In case a computer run is about to terminate, plot results and punch the latest variable values on cards for a subsequent continued run.

(6) Allow conditional MONITOR printout of variables in any one of 12 subprograms for debugging purposes.

(7) Specify the normal conditions and circuit-failure conditions for run termination.

(8) Apply a resistor model (see pp. 15-17) whose resistance is constant or variable with time.

(9) Apply an inductor model (see pp. 20-23) composed of a linear or nonlinear inductance in parallel with a shunt resistance--both in series with a resistance of any value.

The modified TRAC program, referred to as MTRAC, will be described next in three parts. In Part A, the overall organization will be discussed by describing the functions of the various subroutines and blocks in the main program. A more detailed description is given in Part B

for the READ-WRITE subroutines and the computation subroutines of the circuit elements, and in Part C for the main program and a few subroutines associated with it.

A. Program Organization

The overall organization of the MTRAC program is summarized by the functional block diagram in Fig. 28. Blocks 1-5 cover the initialization, and the remaining blocks cover the computation involved at every Δt during the transient time and the output.

1. Initialization

Block 1 - To begin with, the general input data are read in and printed out for each mode of operation.*

Block 2 - The network and parameter data of each circuit element are read in and printed out for each mode of operation.* This operation is performed by calling each of the READ-WRITE circuit-element subroutines, to be described in more detail in Part B, pp. 83-86.

Block 3 - If so specified by the user, the initial conditions (nodal voltages) will be solved for in an iterative fashion. Alternatively, if the initial conditions are zeroed or read in, this block will be bypassed, as is indicated by the dashed line in Fig. 28.

Block 4 - The initial values of the nodal voltages and selected auxiliary variables (branch voltages, currents, F, ϕ, and $\dot{\phi}$ of certain cores, etc.) are printed out. The values of those variables that are to be plotted are stored in a special PLOT array.

Block 5 - If the run is continued from a previous run (which was terminated due to a limit on the run time), the values of the necessary variables from the previous run are read in and printed out; otherwise, this block is bypassed.

---

* This is unfortunate in view of the large portion of the circuit data that is common to successive modes of operation. Future program modification should eliminate this drawback by reading in only *changes* in the input data.

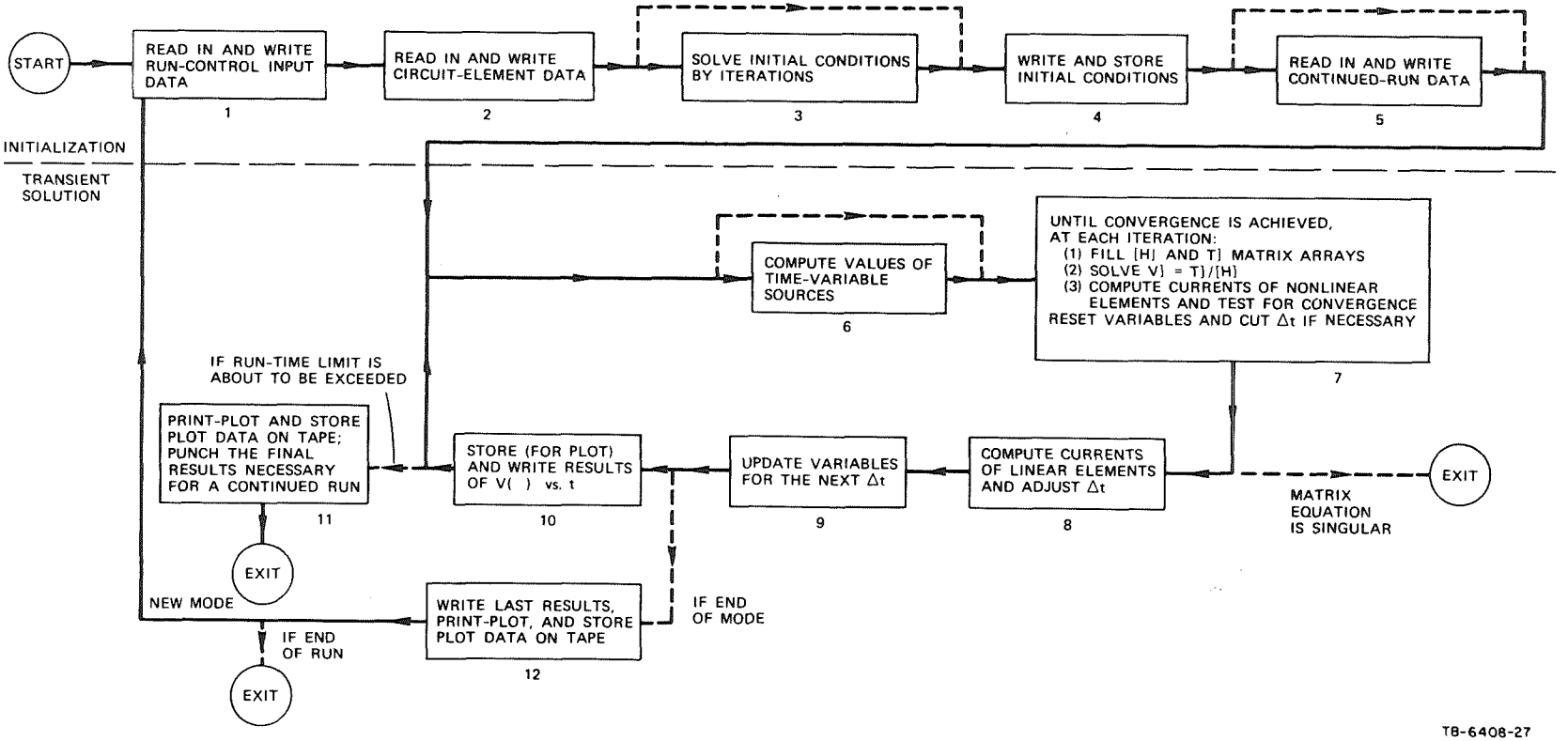


Figure 28 OVERALL ORGANIZATION OF THE MTRAC PROGRAM

2. Transient Computation

Block 6 - The magnitudes of the voltage and current sources that vary with time are computed. If there is no time-variable source, this block is bypassed.

Block 7 - The solution of $V] = T]/[H]$ at a given time step is obtained in an iterative fashion until convergence is achieved. Each iteration consists of three steps:

(1) The arrays of the conductance-matrix $[H]$ and the current-matrix $T]$ are filled by calling the circuit-element subroutines.

(2) The voltage-matrix $V]$ is solved for, using Gaussian elimination and back substitution.

(3) The currents through the nonlinear circuit elements (*i.e.*, diodes, transistors, and magnetic-core windings) are computed and tested for convergence.

If convergence has not been achieved, then these three steps are repeated. Both Δt and the maximum number of iterations vary automatically according to the convergence condition. If the matrix equation is singular, then the contents of the H array are printed out and the run is terminated.

Block 8 - After convergence has been achieved, the currents through the linear elements as well as any auxiliary variables defined by the user are computed. In addition, Δt for the next time step is adjusted according to the number of iterations at the recent convergence.

Block 9 - Time is stepped up by Δt, and the V array is stored in the $V_{(-1)}$ array in preparation for the next Δt. A test for an end of mode is made by comparing t with the mode duration.

Block 10 - If some of the results are to be plotted, then the corresponding V and t values are stored in appropriate arrays, If so specified, the resulting V and t values are printed out. The printout also includes the Δt value, the number of iterations, the index number of the recent time step, and the total (integrated) number of iterations.

Block 11 - If a short time (*e.g.*, 20 seconds) before the specified maximum run time is reached and the run is not completed, then the computed variables to be plotted are print-plotted, the plot data are stored on tape, the final results that are necessary to continue the run

later are punched out on cards, and the run is terminated. The punched final results include the time-step index number, t, Δt, $\Delta t_{(-1)}$, $\Delta t_{(-2)}$, all the V values, capacitor and inductor currents, the values of voltage, current, and $i_f(V)$ slope (see Fig. 17) of each diode and transistor junction, and the values of $F_{(-3)}$, $F_{(-2)}$, $F_{(-1)}$, $\Phi_{(-3)}$, $\Phi_{(-2)}$, $\Phi_{(-1)}$, $\dot{\Phi}_{\epsilon(-1)}$, $\dot{\Phi}_{(-2)}$, $\dot{\Phi}_{(-1)}$, and every winding current of each core.

Block 12 - At the end of a mode, the previous results are printed out as in Block 10. Those variables that have been specified by the user to be plotted are print-plotted, and the stored plot data and plot instructions (frame, tics, scale, units, labels, etc.) are written on a plot magnetic tape.

If a new mode follows, the computation is repeated, starting from Block 1; otherwise, the run is terminated.

B. Circuit-Element Subroutines

Two types of circuit-element subroutines are distinguished:

(1) The READ-WRITE (RW) subroutines that handle the reading in and printing out of the circuit-element input data (they may also compute certain parameters), and

(2) The *computation* subroutines, where the contributions to the H and T arrays are computed and where currents are computed and, for nonlinear elements, tested for convergence.

The names of the RW subroutines and the computation subroutines corresponding to the eight circuit-element types are listed in Table III.

Table III

CIRCUIT-ELEMENT SUBROUTINES

READ-WRITE SUBROUTINE	COMPUTATION SUBROUTINE	CIRCUIT ELEMENTS
RWIVS	IVS	current sources and floating-voltage sources
RWRES	RES	resistors (constant or time-dependent)
RWCAP	CAP	capacitors
RWIND	IND	inductors (constant or current-dependent)
RWZEN	ZEN	zener diodes
RWDIOD	DIOD	diodes
RWTRAN	TRAN	transistors (npn or pnp)
RWCORE and RWCORF or RWCORH	CORE	magnetic cores (ferrite or tape-wound)

In the case of magnetic cores, a short subroutine named RWCORE will, in turn, call either Subroutine RWCORF for ϕF input data or Subroutine RWCORH for BH input data (*cf.* Sec. I-C-3-a, pp. 64-68, and Sec. I-C-3-b, pp. 69-78).

Let us describe each subroutine type separately in a general way, and then describe one subroutine, Subroutine CORE, in more detail.

1. READ-WRITE Circuit-Element Subroutines

The READ-WRITE subroutines are called once (in the beginning of each mode run) from the main program. The tasks of each of these subroutines are to read in and print out the corresponding number of elements, the element type, the circuit data (type and index number of each terminal), and the parameter values of each circuit element. Referring to Fig. 29, let us describe each task separately.

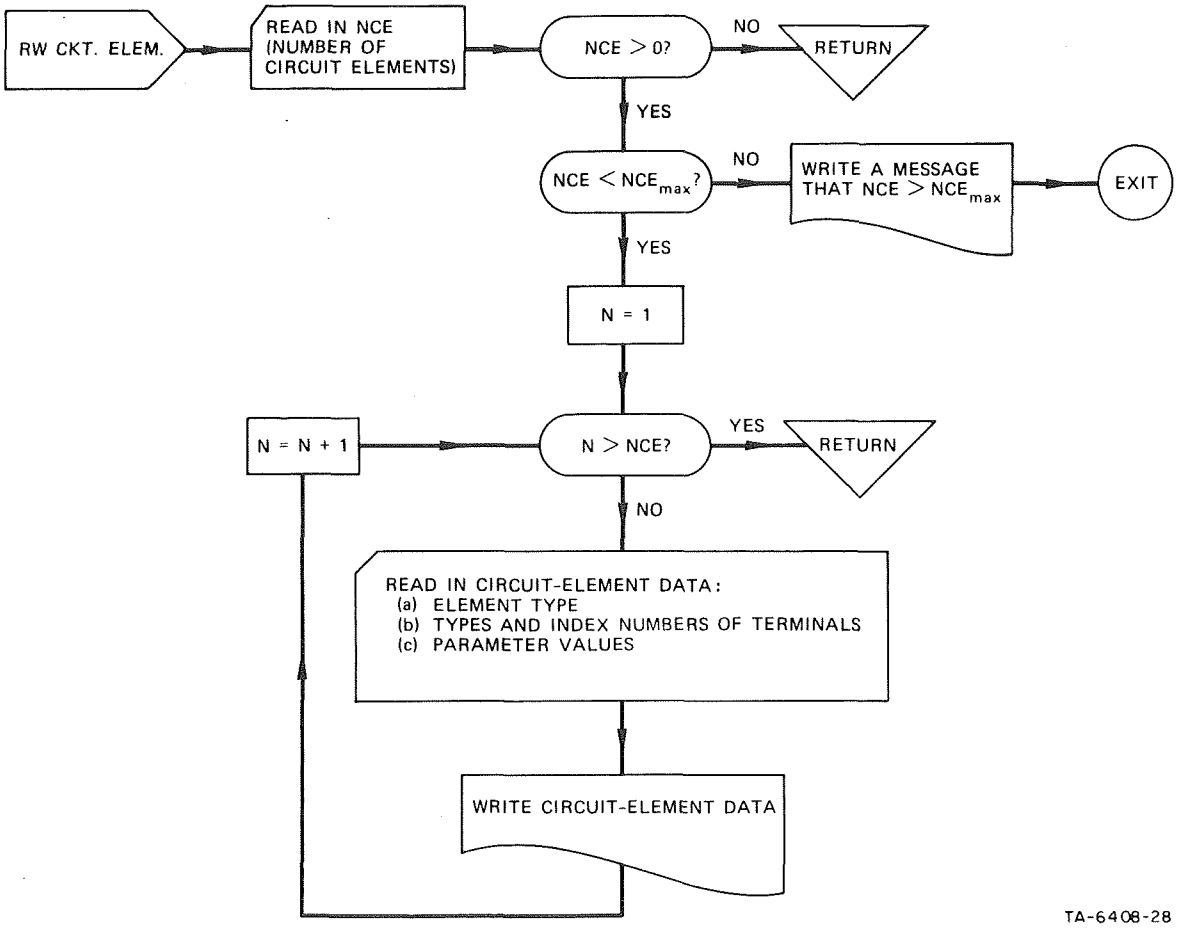


Figure 29 A GENERAL FLOW CHART FOR A READ-WRITE CIRCUIT-ELEMENT SUBROUTINE

a. Number of Circuit Elements

The number of the circuit elements of each of the eight element types, which is denoted in Fig. 29 by NCE for the sake of generality, is read in first. If NCE ≤ 0 (*i.e.*, no element of a given type), then there is a RETURN from the subroutine. On the other hand, if NCE exceeds the maximum allowed number of elements, NCE_{max}, then a message is printed out to this effect, and the run is terminated by calling EXIT. The values of NCE_{max} in MTRAC are listed in Table IV.

Table IV

MAXIMUM NUMBER OF CIRCUIT ELEMENTS

CIRCUIT ELEMENT	SUBROUTINE NAME	NCE NAME	NCE_{max}
Current or floating-voltage sources	RWIVS	NIV	20
Resistors	RWRES	NR	40
Capacitors	RWCAP	NC	20
Inductors	RWIND	NL	20
Zener diodes	RWZEN	NZEN	20
Diodes	RWDIOD	NDIODE	20
Transistors	RWTRAN	NTRANS	20
Cores	RWCORE	NCORE	20
Windings of each core		NW(N)	10

b. Circuit Element Types

Certain circuit elements may have different types. These are listed in Table V together with the values of the corresponding indicators in MTRAC.

c. Types and Index Numbers of Terminals

The types and index numbers of each terminal are read in, using alphanumeric *blank*, *S*, or *G* to designate a floating terminal, a terminal tied to a voltage source, or a terminal tied to ground, respectively. Referring to Fig. 1, Terminal-*a* data are read in first and Terminal-*b* data are read in second, where *a*-to-*b* is referred to as the positive direction of the device current. This direction is arbitrary for a resistor, a capacitor, and an inductor, but is fixed for the remaining circuit elements. The order in which the terminal data are read is listed in Table VI.

Table V

CIRCUIT-ELEMENT TYPES

CIRCUIT ELEMENT	TYPES	INDICATOR NAME AND VALUE
Sources		
Characteristic	(1) Current source (2) Floating-voltage source	K4(N) = 1 K4(N) = 0
vs. time	(1) dc (2) Time variable	K3(N) = 0 K3(N) = No. of variable source
Resistors	(1) Constant (linear) (2) Time variable (nonlinear)	NRP(N) = 0 NRP(N) > 0
Inductors	(1) Constant (linear) (2) Current variable (nonlinear)	VICON(N) = 0 VICON(N) > 0
Transistors	(1) npn (2) pnp	K1(N) = 0 K1(N) = 1
Cores		
Input data	(1) ϕF (2) BH	PARIN = PHIF PARIN = BH
Static $\phi(F)$ or B(H)	*A*; *B*; *C*; *D*; *E*; *F*; or *G*	TYPE (N) = *A*; *B*; *C*; *D*; *E*; *F*; or *G*
Material	(1) Ferrite (2) Tape-wound metal	CORMAT(N) = *F* CORMAT(N) = *T*

Table VI

ORDER OF TERMINAL DATA

CIRCUIT ELEMENT	TERMINAL ORDER *a* - *b*	Fig. No.
Current source	"From" terminal - "To" terminal	7
Floating-voltage source	Positive terminal - Negative terminal	11
Resistor	arbitrary	
Capacitor	arbitrary	
Inductor	arbitrary	
Zener diode	Anode - Cathode	14
Diode	Anode - Cathode	15
Transistor	Base - Collector - Emitter	18
Core winding	Bottom terminal - Top terminal	19

d. Parameter Values

Finally, the data of the circuit-element parameters are read in. These parameters are listed in Table VII.

2. Computation Circuit-Element Subroutines

The eight computation circuit-element subroutines are listed in Table III. These subroutines are called at each iteration by a central subroutine named ELEM. A general functional flow chart for each of the circuit-element subroutines is shown in Fig. 30. The tasks of each of these subroutines are to fill the *H* and *T* arrays and to compute the

Table VII

INPUT DATA OF CIRCUIT-ELEMENT PARAMETERS

CIRCUIT ELEMENT	PARAMETERS	REFERENCE
Current source	Amplitude (if time variable, specify index number of variable source).	
Floating-voltage source	V_s (if time variable, specify index number of variable source); R_s.	Fig. 11
Resistor	R; NRP (Number of resistance-*vs.*-t points, including initial); if NRP $\geq$ 2, then specify also R_f, t_f, and p for each of the NRP $R(t)$ modes; *cf.* Fig. 10, where NRP = 5.	Eqs. (78) through (81); Figs. 9 and 10
Capacitor	C; R_C; R_{CSH}; initial current (optional).	Fig. 12
Inductor	L (linear) or L_0 (nonlinear); R_L; R_{LSH}; $1/I_{con}$ (blank if linear); initial current (optional).	Fig. 13; Eq. (109)
Zener diode	R_z; R_{zb}; E_{zb}.	Fig. 14(c)
Diode	I_{sd}; m_d; $R\ell_d$; C_{j0d}; $V_{\varphi d}$; T_d.	Fig. 15; Eqs. (117) through (121); Eq. (128)
Transistor	β_n; β_i; T_e; T_c; I_{sc}; m_c; C_{j0c}; $V_{\varphi c}$; $R\ell_c$; I_{se}; m_e; C_{j0e}; $V_{\varphi e}$; $R\ell_e$.	Fig. 18
Core		
(1) ϕF Data	n_w; initial ϕ/ϕ_r; *OD* (inch); *ID* (inch); ϕ_s; ϕ_r; $\phi_d(F)$ Type; core material; F_{d1}; F_{dz}; F_{d2}; F_{d3}; ϕ_{d2}; H_a; H_q; H_n; λ_d; ν_d; F_{dB}; F_0''; λ; ν; F_B; F_0; ρ_p; F_{B1}; F_{01}; ρ_{p1}.	Figs. 23 and 24
(2) *BH* Data (B_{in} in gausses; H_{in} in oersted)	n_w; initial B/B_r; *OD* (inch); *ID* (inch); A_{in} (cm^2); $B_{r,in}$; $B_d(H)$ Type; core material; $H_{d1,in}$; $H_{dz,in}$; $H_{d2,in}$; $H_{d3,in}$; $H_{m,in}$; $B_{d2,in}$; $B_{d3,in}$; $B_{m,in}$; n_{dynp}; for $j = 1,2,\ldots n_{dynp}$: $f_{(j)}$ and $H_{OB(j)}$.	Fig. 25

circuit-element currents. Accordingly, each subroutine consists of two major parts. Part (1) in Fig. 30 corresponds to a two-terminal circuit element; for a transistor, Part (1) is similar, except that there are four possible terminal types. Let us describe each part separately, assuming first a transient solution. A solution of initial conditions will be discussed subsequently. Finally, we shall describe Subroutine ELEM.

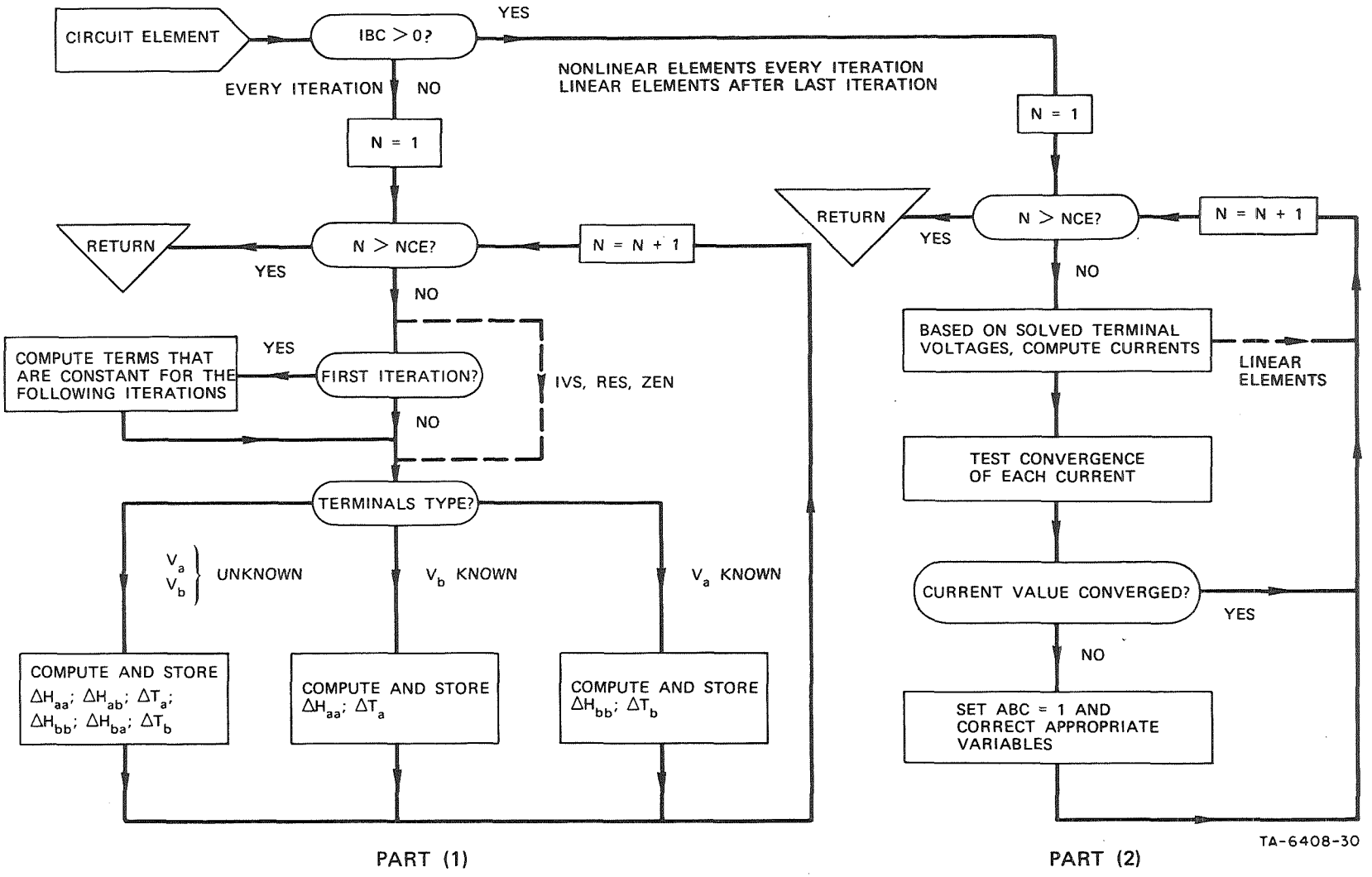


Figure 30 A GENERAL FLOW CHART FOR A TWO-TERMINAL CIRCUIT-ELEMENT SUBROUTINE

a. Part (1) - Filling the H and T Arrays

In the beginning of each iteration, the H and T arrays are cleared (*i.e.*, their elements are set to zero) by an instruction from Subroutine ELEM, and a control indicator named IBC is set to zero. Then Subroutine ELEM calls Subroutine IVS, and the contributions of every current or floating-voltage source to the H and T arrays are computed and stored by executing the first part of Subroutine IVS. This is then repeated for all the resistors, capacitors, inductors, zener diodes, diodes, transistors, and cores by, respectively, calling Subroutine RES, CAP, IND, ZEN, DIOD, TRAN, and CORE.

If we examine the expressions for ΔH and ΔT of capacitors, inductors, diodes, transistors, and cores (see Secs. I-B and I-C), we find that certain terms in these expressions are constant for a given Δt. Therefore, in order to save computation time, these terms will be computed at the first iteration only and saved for the following iterations. The entire expressions for ΔH and ΔT of a capacitor, Eqs. (96) and (97), and an inductor, Eqs. (107) and (108), are under this category. In the diode case, these terms are expressed in Eqs. (145) through (147); these terms will be computed only if there is a change in Δt. (If the small variations in the values of C_{jd}, Eq. (128), for the different iterations were neglected, then additional terms, such as $C_{jd}V_{d(-1)}/\Delta t$, might also be computed only once every Δt.) Constant terms for a transistor, Eqs. (193) through (198), are similarly computed at the first iteration only if there is a change in the value of Δt. In the case of a square-loop core, terms that are constant for a given Δt are $\dot{\phi}_\epsilon$ and $\dot{\phi}'_\epsilon$, Eqs. (253) and (254). No such preliminary computation is needed for sources, resistors, and zener diodes, as is indicated by the dashed line in Part (1) of Fig. 30.

We have seen in Sec. I-A-2 that three cases are distinguished for a two-terminal circuit element. Accordingly, the terminals' type is tested next. If both V_a and V_b are unknown voltages, then ΔH_{aa}, ΔH_{ab}, ΔT_a, ΔH_{bb}, ΔH_{ba}, and ΔT_b are computed from Eqs. (17), (18), (19), (23), (24), and (25), respectively, and stored in the corresponding locations in the H and T arrays. If no more than one circuit element is connected across Nodes a and b, then $\Delta H_{ab} = H_{ab}$ and $\Delta H_{ba} = H_{ba}$. If V_b is known (source or ground), then only ΔH_{aa} and ΔT_a, Eqs. (28) and (29), are computed and stored. Finally, if V_a is known (source or ground),

then only ΔH_{bb} and ΔT_b, Eqs. (32) and (33), are computed and stored. The flow chart for Part (1) of the transistor subroutine TRAN is similar to the one in Fig. 30, except that there are four terminal types to choose from (see Fig. 6).

The expressions for ΔH and ΔT appearing in Eqs. (17) through (33) are, of course, different for each circuit element. These expressions are given in the following equations:

Current source	Eqs. (73) and (74)
Voltage source	Eqs. (83) and (84)
Resistor	Eqs. (76) through (78)
Capacitor	Eqs. (88) through (97)
Inductor	Eqs. (102) through (110)
Zener diode	Eqs. (114) and (115)
Diode	Eqs. (141) through (152)
Transistor	Eqs. (183) through (212)
Core	Eqs. (219) through (236), (240) through (263), and (270) through (309).

b. Part (2) - Computation of Currents and Convergence

After returning from Part (1) of the last circuit-element subroutine (CORE), the main program calls a matrix-equation subroutine, named SEQSOL, which solves $V] = T]/[H]$. Subroutine ELEM now sets IBC to 1 and a convergence indicator named ABC to -1, and calls Part (2) of each circuit-element subroutine, whose flow chart is also shown in Fig. 30. Here, the element subroutines are classified into two groups, depending on whether the circuit element is linear or nonlinear.

The linear-element subroutines are IVS, RES, CAP, IND, and ZEN (a zener diode is treated as a piecewise-linear element in a wide range of voltage, and hence is included in the linear group; a nonlinear inductor is piecewise linearized every time step). In each of these subroutines, the currents are computed at the end of the last iteration, *i.e., after* convergence has been achieved.

At every iteration, Subroutine ELEM calls Parts (2) of the nonlinear-element subroutines (DIOD, TRAN, and CORE), where the current of every diode, transistor junction, and core winding is computed and tested for convergence. If the current of at least one nonlinear circuit element has not converged, then ABC is set to 1, the corresponding current or voltage values are corrected, and Subroutine ELEM begins a new iteration by setting IBC = 0 and calling Part (1) of every circuit-element subroutine. When convergence is finally achieved, Subroutine ELEM calls Parts (2) of the linear-element subroutines (IVS, RES, CAP, IND, and ZEN), where the corresponding currents are computed.

Let us now examine the convergence procedure, beginning with the diode and referring to Fig. 17. Assuming that $R_d = 0$, then V_d is computed as a difference between the diode nodal voltages. For each diode, an approximate value of i_{fd} is computed from Eq. (123) and compared with $I_{sd}\,[\exp\,(V_d/\theta_{md}) - 1]$, Eq. (117). If the difference between the two current values is more than 10 percent of the latter, then convergence has not been achieved; in this case, ABC is set to 1 and the values of V_d and s are corrected on the basis of Eqs. (120) and (124), respectively. See Fig. 16. The previous values of i_{fd}, V_d, and s of a given diode remain fixed as long as the convergence criterion for that diode is satisfied.

A computation similar to the above is performed for the base-collector and base-emitter junctions of each transistor.

The current i of each core winding is computed from Eqs. (218) through (220), which include the Newton-Raphson convergence method. The latter is now modified if the sign of $(i - i_{<-1>})$ is opposite to the sign of $(i_{<-1>} - i_{<-2>})$ (*i.e.*, i is oscillatory) in order to speed up the convergence. This modification may be performed in one of two ways:

(1) Every other iteration, i is corrected on the basis of Aitken's formula[20]

$$i_{(new)} = i - \frac{(i - i_{<-1>})^2}{i - 2i_{<-1>} + i_{<-2>}} . \qquad (362)$$

(2) Every iteration, i is corrected on the basis of the simple formula[3-5]

$$i_{(new)} = 0.5(i + i_{<-1>}) \quad . \tag{363}$$

Although our experience so far has not shown which formula is superior, Eq. (362) will normally be used because it is more powerful. However, the user may choose to use Eq. (363) instead. The value of i, whether corrected by one of the above formulas or not, is then compared with $i_{<-1>}$. Convergence is achieved if

$$|i - i_{<-1>}| \leq \epsilon_{rel} |i| + \epsilon_{abs}, \tag{364}$$

where ϵ_{rel} and ϵ_{abs} are the relative and absolute errors, respectively (for example, $\epsilon_{rel} = 10^{-3}$ and $\epsilon_{abs} = 10^{-5}$ ampere). If convergence is not achieved for any winding current, then ABC is set to 1. If convergence of a given winding current is achieved, then, in order to cut down the number of iterations, i is modified according to

$$i_{(new)} = |p| i_{<-1>} + (1 - |p|) i \quad , \tag{365}$$

where $|p|$ is a factor varying from 0 to 1, and is specified by the user. If $p = 0$, then $i_{(new)} = i$ (*i.e.*, no modification), but if $|p| = 1$, then $i_{(new)} = i_{<-1>}$ (*i.e.*, the change in i at the last iteration is disregarded). Furthermore, if Eq. (363) is to be used instead of Eq. (362), then a negative p will be specified. (If, in addition, p should be zero, then p will be set to a small value, *e.g.*, $p = -1.0 \cdot 10^{-30}$.)

c. Parts (1) and (2) for Solution of Initial Conditions

Parts (1) and (2) of Subroutines IVS, RES, and ZEN for either a transient solution or a solution of initial conditions are identical. On the other hand, the presence of Δt in the transient-solution expressions for ΔH and ΔT of the remaining circuit elements requires modifications of Parts (1) and (2) when solving for initial conditions. Here we distinguish between capacitive elements and inductive elements.

The capacitive circuit elements include capacitors, diodes, and transistors. In an initial dc state, the capacitance is characterized by no current flow, and thus, following Eq. (97), the corresponding ΔH and

ΔT should both be zero if $R_{CSH} = \infty$ (see Fig. 12). Referring to Eqs. (88) through (97) for a capacitor, we observe that, since $i_{ab(-1)} = 0$, this can be achieved by simply letting $\Delta t \rightarrow \infty$. In MTRAC, this condition is simulated by letting $\Delta t = 10^{20}$ when solving for the initial conditions. Referring to Fig. 15, a static diode model is similarly achieved by eliminating both the junction capacitance C_{jd} and the diffusion capacitance represented by the current source $T_d(di_{fd}/dt)$. Referring to Eqs. (131) and (132), this condition, too, is achieved by letting $\Delta t = 10^{20}$. A similar condition is achieved for each junction of every transistor. [*cf.* Eqs. (156) through (167)].

Let us now consider the initial conditions of inductive elements (inductors and cores). Two cases are distinguished for an inductor:

First, a dc case in which $i \equiv i_L$ is solved for. Referring to Fig. 13, in this case $di_1/dt = 0$, and an inductor model is reduced to a simple resistor, $R = R_L$. Therefore, following Eqs. (76) and (77), $\Delta H = 1/R_L$ and $\Delta T = 0$. After the solution of the initial condition has converged, the initial current through the inductor is computed, using Eq. (75), *i.e.*, $i_L = (V_a - V_b)/R_L$.

Second, a dc case in which the initial value of i_L is given. This case may occur when either the initial current, i_{L0}, is known to the user [a variable INCIL(N) of the Nth inductor is then set to 1 and and i_L is set equal to the read-in value of i_{L0}] or when the initial current is equal to the final current solved at a previous mode of operation (a variable NSVIL is then set to 1, and i_L of every inductor is saved from the previous mode). In this case, the inductor is represented by a current source, $i_s = i_L$. Therefore, following Eqs. (73) and (74), $\Delta H = 0$ and $\Delta T = -i_L$. Both cases are treated in a special part for solution of initial conditions in Subroutine IND.

Consider now a magnetic core in a static state. Since $\dot{\phi} = 0$, each jth core winding is represented by a simple resistor, $R = R_{wj}$, where R_{wj} is the inherent winding resistance of the jth winding, Fig. 19. Therefore, $\Delta H_j = 1/R_{wj}$ and $\Delta T_j = 0$ and, after convergence has been achieved, the jth initial current is computed from the relation $i_j = (V_a - V_b)/R_{wj}$. Furthermore, the initial MMF of each core is also computed, using the relation $F = \sum_j N_j i_j$.

d. Subroutine ELEM

Subroutine ELEM (listed on pp. 160-161 of Vol. II) has two major functions:

(1) Alternately calling Parts (1) and (2) of each of the circuit-element subroutines every iteration, and

(2) Controlling the means of convergence by
   (a) Adjusting the magnitude of Δt
   (b) Adjusting the maximum number of iterations
   (c) Resetting time, nodal voltages, and iteration count to their initial values (at the beginning of the first iteration) if the number of iterations exceeds its maximum value.

Before describing Subroutine ELEM, let us define eleven variables in this subroutine:

ABC — Convergence indicator:
1 - Convergence has not been achieved; 0 - otherwise.

IBC — Direction indicator in a circuit-element subroutine:
0 - to Part (1) (filling the H and T arrays);
1 - to Part (2) (computing currents and, if the element is nonlinear, testing convergence).

ITRMAX — The maximum number of iterations above which Δt is cut by a factor of 4 and the iterative computation is restarted by resetting the nodal voltages and other variables to their initial values at the first iteration.

ITRTST — The number of iterations (at the previous time step) above which Δt is halved and below which Δt is doubled (provided that Δt does not exceed its maximum value, TI, which is specified by the user).

KFAIL — The number of variable resettings (and Δt cutting by a factor of 4) without succeeding to converge, using a fixed ITRMAX value.

KFMAX — The maximum value of KFAIL above which ITRMAX is doubled.

KONVRG — The number of times convergence has been achieved within less than ITRTST iterations.

NITER — The number of iterations.

MAXMIN — The minimum (and initial) value of ITRMAX.

NDELT — Index number of the time step Δt.

TI — The maximum value of Δt (specified by the user).

A functional flow chart of Subroutine ELEM is shown in Fig. 31. This subroutine is called by the main program twice every iteration, and performs the following tasks:

First, the H and T arrays are cleared and then filled by calling a subroutine named AUX(K) (with $K = 0$) and all the circuit-element subroutines. Subroutine AUX(K) is written by the user in the FORTRAN-IV language and constitutes a source deck. This allows the user to modify Part (1) of any element (in which case, $K = 0$) and/or to define any auxiliary output variable to be computed after convergence has been achieved in Part (2) (in which case, $K = 1$).

Second, the current of every diode, transistor junction, and core winding is computed and tested for convergence by calling Parts (2) of Subroutines DIOD, TRAN, and CORE. The iterative transient solution is controlled automatically by varying Δt and the maximum number of iterations in a manner described as follows.

Refer to Fig. 31. Initially, ITRMAX is set equal to MAXMIN (*e.g.*, 20), which may be specified by the user. If convergence is not achieved within ITRMAX iterations, then KFAIL is increased by one and there are two possibilities, depending on KFAIL:

(1) If KFAIL is smaller than KFMAX (*e.g.*, 3), then Δt is cut by a factor of 4, and the iterative computation is restarted by resetting the time and all the pertinent variables to their initial values at the first iteration.

(2) If KFAIL reaches KFMAX, then the upper limit of iterations, ITRMAX, is doubled, KFAIL is reset to zero, and the above iterative process is repeated.

If convergence is achieved, then the following steps are followed:

(1) Subroutines IVS, RES, CAP, IND, and ZEN are called in order to compute the corresponding device currents, and Subroutine AUX(1) is called in order to set or compute the auxiliary variables (*e.g.*, voltage differences, circuit-element currents, F, ϕ, and $\dot{\phi}$ of cores, or any legitimate function defined by the user).

(2) Updating $\Delta t_{(-2)} = \Delta t_{(-1)}$, $\Delta t_{(-1)} = \Delta t$, and adding 1 to NDELT.

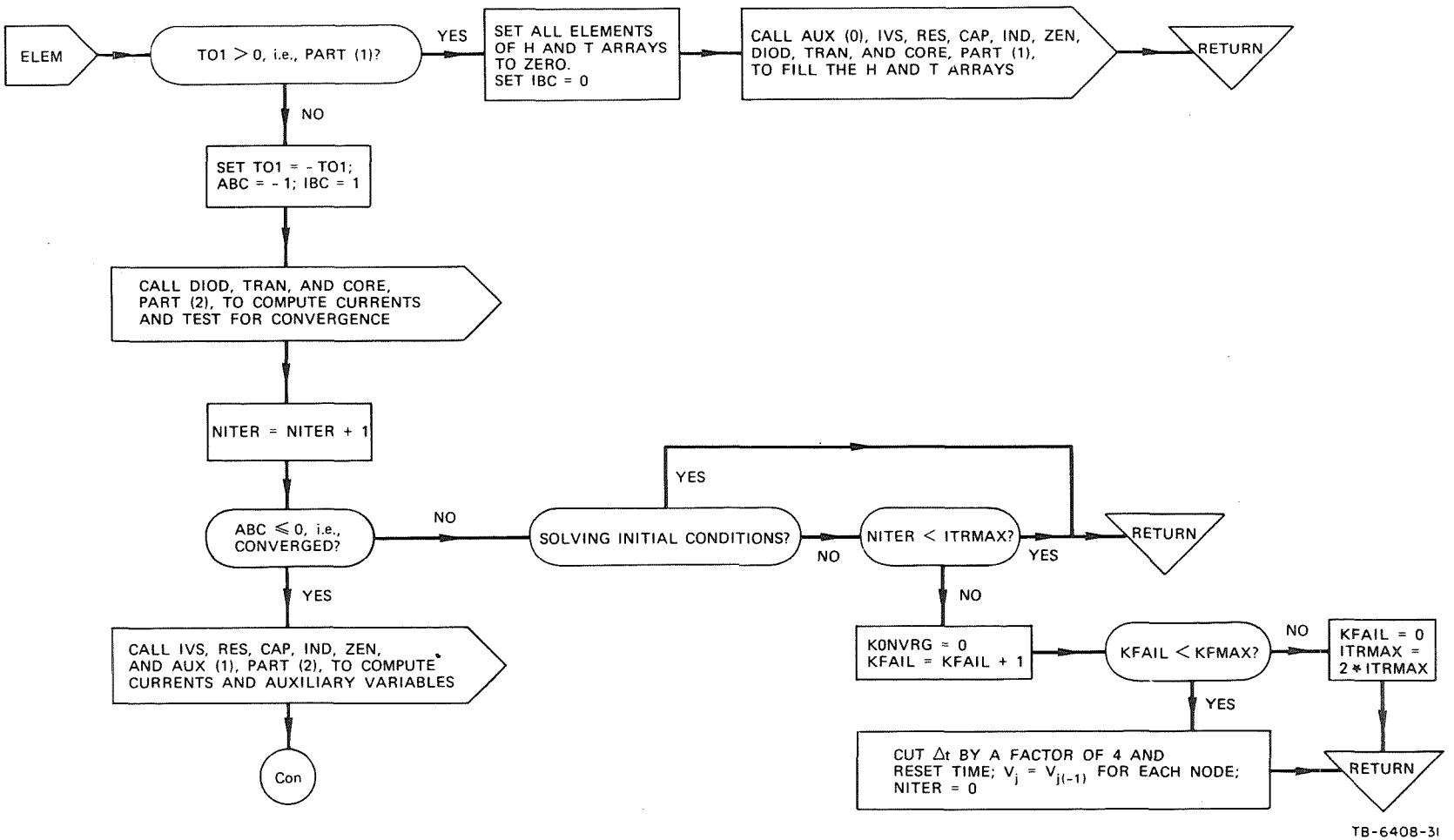


Figure 31 A FUNCTIONAL FLOW CHART OF SUBROUTINE ELEM

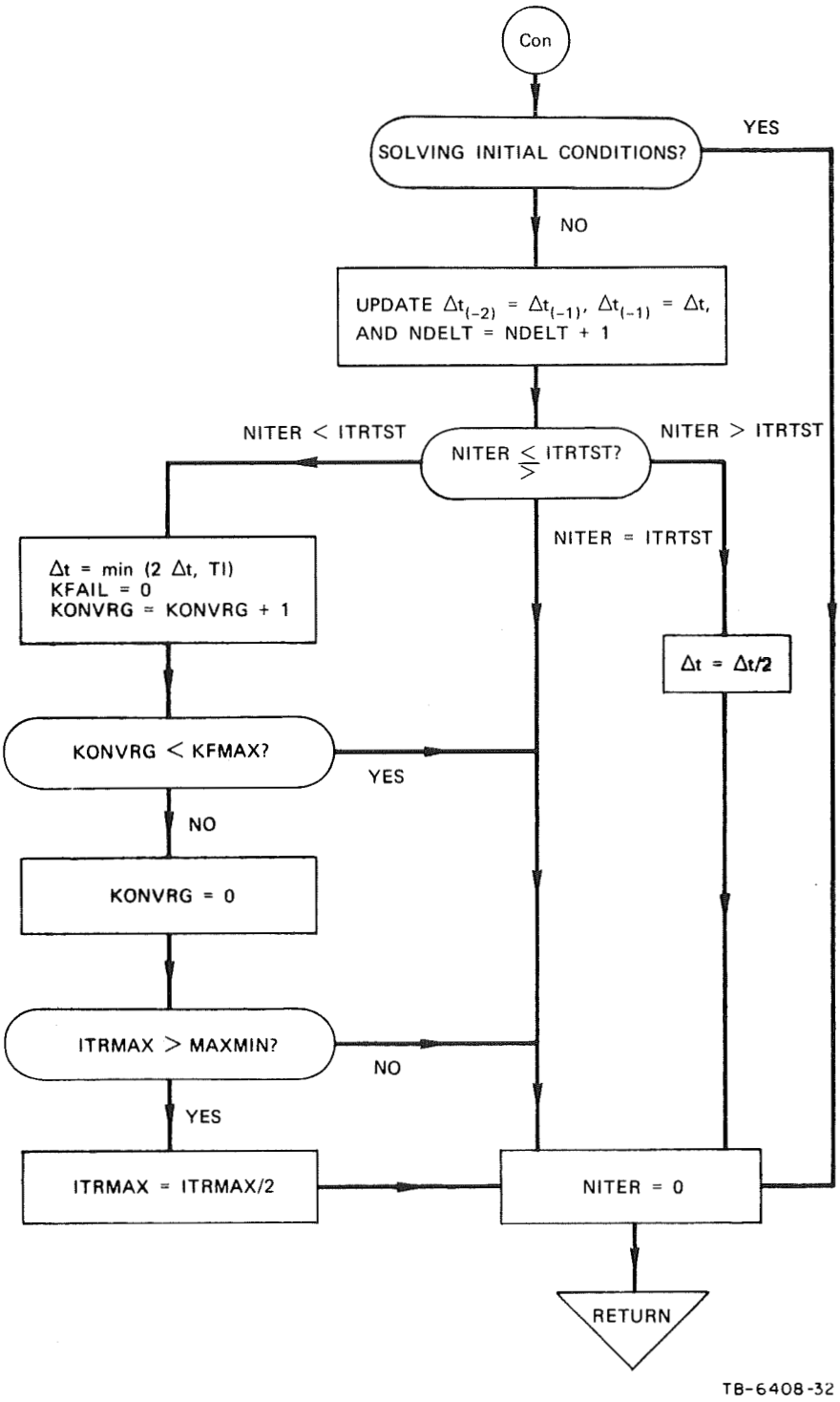


Figure 31 Concluded

(3) The value of Δt for the following time step may be changed according to the number of iterations, NITER, corresponding to the previous convergence:

(a) If NITER > ITRTST, then achieving convergence is regarded "hard", and Δt is cut by a factor of 2.

(b) If NITER = ITRTST, then there is no change in Δt.

(c) If NITER < ITRTST, then achieving convergence is regarded "easy", and the following actions are performed:

(*i*) The time step Δt is doubled, provided it does not exceed the maximum value, TI, specified by the user.

(*ii*) KFAIL is set to zero and KONVRG is increased by 1. If the number of successive convergence successes, KONVRG, reaches KFMAX, then KONVRG is set to zero and ITRMAX is cut by a factor of 2, provided that it is larger than MAXMIN.

The above automatic adjustment of Δt and ITRMAX decreases the time step and increases the allowed number of iterations if convergence is hard to achieve. Subsequently, if convergence becomes easy to achieve, then Δt is restored to a larger value and ITRMAX is restored to a smaller value.

3. Magnetic-Core Subroutines

The READ-WRITE subroutine and the computation subroutine of one of the circuit elements will now be given as an example. We choose the magnetic-core subroutines because they are the main product of this project and because they are larger and more complex than the other subroutines.

Most of the variables in the core subroutines correspond phonetically to the variables in Sec. I-C. These variables are defined in Sec. II-A of Vol. II of this report.

a. Subroutines RWCORE, RWCORF, and RWCORH

Subroutine RWCORE reads in the number of cores, NCORE, and the kind of parameter input data, PARIN. If NCORE > 0, then Subroutine RWCORE calls one of two subroutines, depending on PARIN:

(1) If PARIN = PHIF or blank, then Subroutine RWCORF is called for reading in and processing $\phi_d(F)$ and $\dot{\phi}_p(F)$ data ("ϕF" data),

(2) If PARIN = BH, then Subroutine RWCORH is called for reading in and processing static and dynamic B(H)-loop data (usually of tape-wound cores) found in a manufacturer's catalog.

Referring to the listings of Subroutines RWCORF and RWCORH on pp. 137-151 of Volume II, each of these two subroutines is described briefly as follows.

i. Subroutine RWCORF

First, Subroutine RWCORF reads in general data: relative and absolute errors for convergence criteria of winding currents, whether or not to terminate the run upon completion of the flux switching in all cores, and whether or not to include $\dot{\phi}_\epsilon$ in $\dot{\phi}$.

If ϕ of each core is saved from the previous mode (NSVPHI = 1), then, unless all variables are saved (NSVAR = 1) or the run is continued from a previous time-limited run (KONT = 1), the value of the remanent flux is computed by subtracting the elastic $|\Delta\phi_\epsilon|$ corresponding to F at the end of the previous mode from ϕ if $F > 0$, or adding $|\Delta\phi_\epsilon|$ to ϕ if $F < 0$. The magnitude of the resulting remanent ϕ should not exceed ϕ_r. (See Loop 7759, *i.e.*, the loop associated with Statement Number 7759.)

Next, Subroutine RWCORF reads in for each core the number of windings, the index number KC1 of the core whose parameters may be copied, and the initial value of ϕ/ϕ_r. If the number of windings is zero or larger than 10, then a message to this effect is printed out and the run is terminated.

The types and index numbers of each core winding are then read in (see Loop 13). This is followed by reading in the static and dynamic core parameters (if KC1 $\leq$ 0) or copying these parameters from CORE No. KC1 (if KC1 > 0).

If DMAX $\neq$ 0, then the worst-case signs ESSES(I) of the 24 parameters of each Nth core are read in, and the corresponding input parameters are modified accordingly.

The auxiliary parameters $p_1(N)$ through $p_{20}(N)$ of each Nth core are computed next. The computation steps follow the procedure outlined in Sec. I-C-3-a, pp. 64-68.

Finally, the circuit and parameter input data as well as the computed auxiliary parameters are printed out.

ii. Subroutine RWCORH

Subroutine RWCORH is similar to Subroutine RWCORF except for the input parameters and the addition of preliminary computation, as described in Sec. I-C-3-b, pp. 69-78.

b. Subroutine CORE

Subroutine CORE includes a section for the solution of initial conditions, in addition to Parts (1) and (2). Referring to the listing of Subroutine CORE on pp. 176-182 of Vol. II, let us discuss each section separately.

i. Initial Conditions

In the solution of the dc initial conditions, an Mth winding of an Nth core is represented by a simple resistance R = RW(N,M). At each iteration, if IBC ≤ 0, the contribution of every RW(N,M) to the H and T matrix arrays is computed.

When convergence is achieved (*i.e.*, when IBC > 0, ABC ≤ 0, and JKKJ ≤ 0), the winding currents are computed. This is followed by computing the initial values of $F = \sum_j N_j i_j$ of each core and setting $F_{(-2)} = F_{(-1)} = F$ and $\phi_{(-1)} = \phi = 0$. If NSVPHI = 0 (*i.e.*, first mode of operation), then ϕ of each core is set equal to the read-in initial flux ϕ_0. The elastic $|\Delta\phi_\epsilon|$ corresponding to F is added to the remanent ϕ if $F > 0$, or subtracted from the remanent ϕ if $F < 0$. In any case, $\phi_{(-1)}$ and $\phi_{(-2)}$ are set equal to the resulting ϕ. Except for computation of the winding currents, this portion of Subroutine CORE is also entered via Statement No. 26 from the main program when the initial variables are either set to zero (NG = 0) or read in (NG = 1).

ii. Part (1)

In addition to the listing of Subroutine CORE on pp. 176-182, refer to pp. 61-90 of Vol. II for the meaning of each variable name in the program (see also Fig. 30).

At the first iteration (when ABC ≤ 0), the values of TM1, TM2, TM3, SUMT, DELTAV, SUMTT, SMTSMT, and SUMTSQ are computed, using Eqs. (240) through (247). For each Nth core, this is followed by compuing SUMF, SUMFT, and SMFTN, following Eqs. (248) through (250). These values are then used to compute the average value of $\dot{F}$, FDOTAV, and the initial guess for F from the three previous results of $F(N)$ *vs.* t, Eqs. (251) and (252). Modifying Eq. (141) in Report 5, the value of ϕ of each core is guessed on the basis of the relation

$$\phi \approx \phi_{(-1)} + \frac{\Delta t}{\Delta t_{(-1)}} [\phi_{(-2)} + 2\Delta t_{(-1)}\dot{\phi}_{(-1)} - \phi_{(-1)}] \quad ,$$

or

$$\phi \approx \phi_{(-1)} + \frac{\Delta t}{\Delta t_{(-1)}} [\phi_{(-2)} - \phi_{(-1)}] + 2\Delta t\dot{\phi}_{(-1)} \quad . \qquad (366)$$

The values of $i_{<-2>}$ and $i_{<-1>}$ of each core winding are set equal to the final solution at the previous Δt.

For each core at each iteration, iterative variables are updated and the computation proceeds after changing the signs of $F(N)$ and PHI(N) if $F(N) < 0$ (see Sec. I-C-2-b, pp. 50-51).

If LASTIC > 0, then PHDTE(N) and PHDTEP(N) (*i.e.*, $\dot{\phi}_\epsilon$ and $\dot{\phi}'_\epsilon$) are computed on the basis of Eqs. (253) through (258) at the first iteration, and remain unchanged during the following iterations. However, during the first three time steps (NDELT ≤ 3), PHDTE(N) and PHDTEP(N) are computed from Eqs. (234) and (235) every iteration. If LASTIC = 0, then the entire computation of PHDTE(N) and PHDTEP(N) is bypassed.

Next, the values of PHID and PHIDP (*i.e.*, ϕ_d and ϕ'_d) are computed on the basis of the 5-region $\phi_d(F)$ model, Eqs. (291) through (304). Computation of PHDTP and PHDTPP on the basis of the $\dot{\phi}_p(F)$ model, Eqs. (308) and (309), then follows. Using Eqs. (262) and (263), the

inelastic PHDTI and PHDTIP (*i.e.*, $\dot{\phi}_i$ and $\dot{\phi}'_i$) are computed and added to their elastic counterparts. The total PHIDOT(N) is then used to compute PHI(N) by employing the trapezoidal integration method. If $\phi > \phi_d$, then ϕ_{ds} is computed from Eqs. (305) and (306) and, if $\phi_d > \phi_{ds}$, then ϕ_d is set equal to ϕ_{ds}.

Finally, following Eqs. (219) through (222), the values of GCNM and VCNM (*i.e.*, G_c and V_c) of each core winding and its contributions to the H and T matrix arrays are computed.

iii. Part (2)

The current of each core winding, CIW, is first computed from Eq. (218) and then, if it appears to be oscillatory, it is corrected by Aitken's method, Eq. (362), every other iteration [or by the "half" method, Eq. (363), every iteration]. If CIW is not close enough to CIJM1(N,M) [the value of CICORE(N,M) at the previous iteration] in accordance with Eq. (364), then convergence has not been achieved (ABC = 1); otherwise, CICORE(N,M) is determined from CIW and CIJM1(N,M), depending on PSTEP (the weighting factor p of the previous iteration), using Eq. (365). If ABC $\leq$ 0, then $F_{(-3)}$, $F_{(-2)}$, $F_{(-1)}$, $\dot{\phi}_{(-2)}$, $\dot{\phi}_{(-1)}$, $\dot{\phi}_{\epsilon(-1)}$, $\phi_{(-3)}$, $\phi_{(-2)}$, and $\phi_{(-1)}$ of each core are updated for the next time step. If the user specified STSWEX = 1 and the flux switching of every core has been completed (FLXSW = 0), then, if NDELT > NUDT, the run will be terminated by setting JCAT = 10 (see description of the main program).

iv. MONITOR WRITE Statements

Twenty conditional WRITE statements are scattered throughout Subroutine CORE for *debugging* purposes. These WRITE statements will be executed if MONCOR = 1. The setting and resetting of MONCOR (and similar logical variables in other subroutines) to TRUE and FALSE are done in the main program, and are controlled by the user, as will be described later. When these WRITE statements are executed, both the names and the values of the following variables are printed out:

Initial Conditions

1 CORES (heading)

2 H(KO1,KO1), H(KO1,KO2), H(KO2,KO1), H(KO2,KO2), T(KO1), T(KO2)

3 CICORE(N,M)

4 CIJM1(N,M), FMM

5 NWSUBN, PHI(N)

Part (1)

6 TM1, TM2, TM3, SUMT, SUMTT, SMTSMT, SUMTSQ

7 CORE N

8 FM3(N), FM2(N), FM1(N), PHDTM2(N), PHDTM1(N), PHIM3(N), PHIM2(N), PHIM1(N), PDTEM1(N), (CICOR1(N,J),J = 1,NWSUBN)

9 SUMF, SUMFT, SMFTN, FDOTAV, F(N), PHI(N)

10 NITER, SIGNF, SF, SPHI

11 EPS, EPSP, FDOTAV, PHDTEP(N), PHDTE(N)

12 Q1, Q2, Q4, Q5, Q6, PHID(N), PHIDP

13 PHDTP, PHDTPP, ETA, PHDTI, PHDTIP, PHIDOT(N), PHDTPR, SPHI, PHIDS, PHI(N)

14 GCNM, VCNM, KO, KO1, KO2

15 H(KO1,KO1), H(KO1,KO2), H(KO2,KO1), H(KO2,KO2), T(KO1), T(KO2)

Part (2)

16 CORE N

17 BV, CIW, CIJM1(N,M), CIJM2(N,M)

18 CIW

19 DID NOT CONVERGE

20 N, M, KO1, KO2, CIW, CICORE(N,M), F(N), ABC.

C. Main Program and Associated Subroutines

The main program and its associated subroutines in MTRAC are described in the following two sections. In Sec. C-1, we outline the main-program steps, including the subroutine calls. In Sec. C-2, we outline some of the subroutines called by the main program. Some Program-Statement Numbers

will be added on the left side of each subprogram outline for orientation in the program listings in Vol. II. The glossary of the variable names is given in Sec. II-A, pp. 61-90, in Vol. II.

1. MTRAC Main Program

The main program calls several subroutines which, in turn, call other subroutines. The subroutine calls, some of which are conditional, are shown in Fig. 32. A conditional call is designated in Fig. 32 by a dashed line, and the condition itself is enclosed in parentheses.

Referring to the glossary of variables (Vol. II, pp. 61-90) and the listing of the main program (Vol. II, pp. 91-99), the main program is outlined as follows.

a. Initialization

Statement No.		
	(1)	Call Subroutine RWRUNC for READ-WRITE of the run-control data.
	(2)	Call all the READ-WRITE circuit-element subroutines.
1116	(3)	Compute the total number of nonlinear elements KNL. If ITRTST, MAXMIN, or KFMAX has not been specified, then set ITRTST = 10 + KNL, MAXMIN = 2(10 + KNL), or KFMAX = 3.
	(4)	Initialize NAW = IBC = JSJ = 0. If KONT = 1 or NSVVAR = 1, go to 2102.
	(5)	Call Subroutines DIOD and TRAN to compute initial currents, and if NG $\neq$ -1, call Subroutine CORE to compute initial values of F, $\dot{\phi}$, and ϕ of each core.
2102		Set JSJ = 1 and KDTIM = 0.
7711	(6)	Call Subroutine INTIAL (see later, pp. 113-114) if solution of initial conditions is specified (NG = -1), but skip this call if the run is continued (KONT = 1) regardless of NG.
1702	(7)	Store and print out the initial values of V1(J) for J = 1 to NZ. Set NG = 0.
	(8)	Start a new run if only solution of initial conditions is specified (N3B < 0) or if no
2601		convergence has been achieved and N3B = 0.

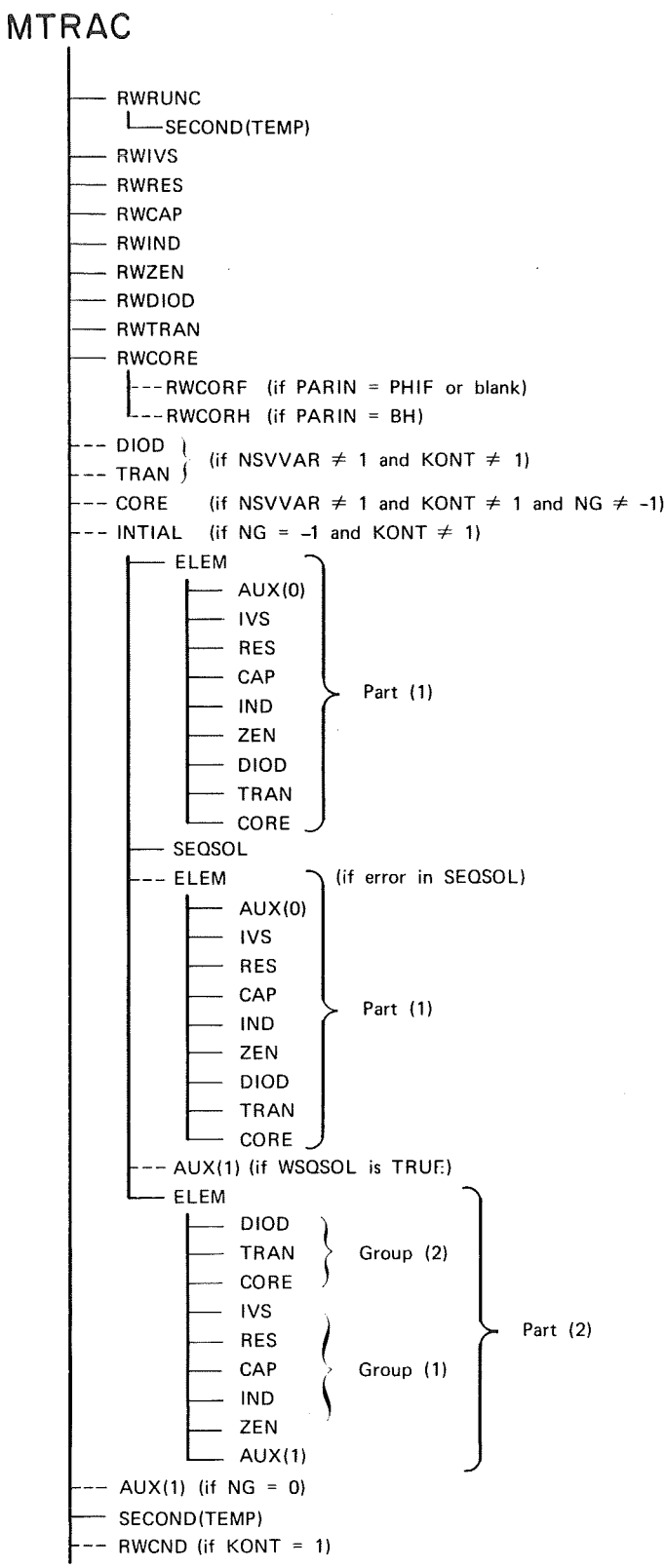


Figure 32 SUBROUTINE CALLS IN MTRAC

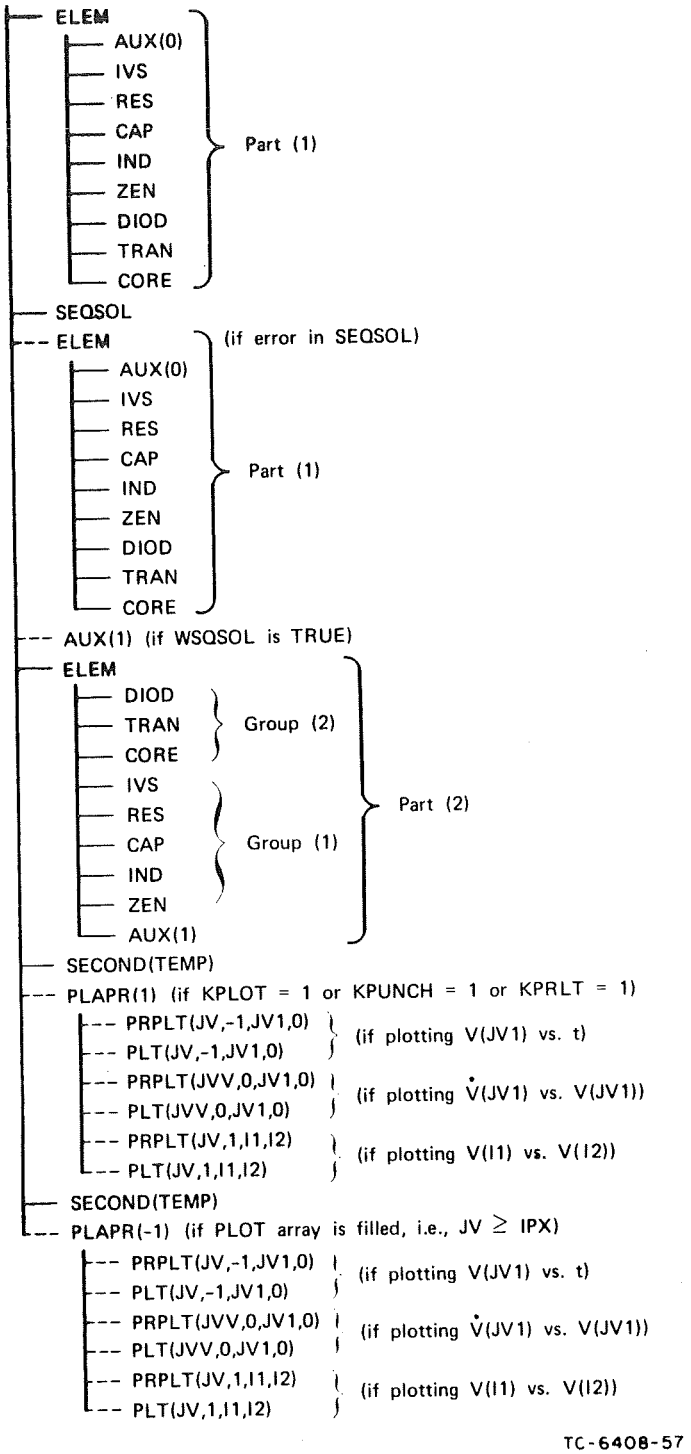


Figure 32 Concluded

2600 (9) If the run is continued (KONT = 1), call Subroutine RWCND (see later p. 114).

2301 (10) Initialize Δt and if NSVVAR $\neq$ 1, set $\Delta t_{(-2)} = \Delta t_{(-1)} = \Delta t$. Store the initial variable values in PLOT array VP(10,350). Set t and store in PLOT array TF(354). Set JSJ = -1.

b. Transient Solution

Transient results are stored in PLOT array VP(10,350) every NB time steps (following the JJ loop) and printed out every NBB · NB time steps (following the JJJ loop). For each time step, do the following.

Statement No.

2222 (1) Set NAW = 0; set NITER = 1.

(2) If there are time-variable sources, then:

(a) Find whether any source has a break point within the time step. If there is a break, cut Δt so that t coincides with the nearest break point among the sources.

3000 (b) For each Jth time-variable source:

(i) If t has not exceeded a break point, compute a new source magnitude, EI(K), using the old slope, E2(J). (Note: K = J - 1, except that K = 19 for J = 1.)

8505 (ii) If t has exceeded a break point, start a new source submode by increasing JG(J) by 1. Compare JG(J) with NI = ST(J,1)(the number of source break points):

If JG(J) < NI, compute new slope and source magnitude.

1071 If JG(J) $\geq$ NI, then

1074 If ST(J,2) = 0 (last value to be held), convert the source to a dc source by setting ST(J,2) = -1 and fixing the source amplitude.

1081 If ST(J,2) > 0 (cycle), reset JG(J) to 1 and reset the source slope and magnitude to their initial values.

119 (3) Test and set the MONITOR printout conditions. At the first time-step for which NDTMIN < NDELT, turn on the conditions to monitor in the subroutines specified by the user and print out these specifications. At the first time that NDTMAX < NDELT, turn off the conditions to monitor.

(4) Iterate in a loop, until convergence is achieved, along the following steps:

701 (a) Call Subroutine ELEM, Part (1), to fill the H and T arrays.

9992 (b) Call Subroutine SEQSOL(ERR) to solve $V] = T]/[H]$.
If the solution fails, call Subroutine ELEM to fill Array H, print out the H values, and terminate the run.

2008 (c) Call Subroutine ELEM, Part (2), to compute currents and test convergence.

(d) Increase NITER by 1 and NITRSM by 1.

(e) If convergence has been achieved (ABC = 0) or has not been achieved within ITRMAX iterations (ABC = -3), get out of the loop. Otherwise, set NAW = -1 if NAW = 0 and start a new iteration (go to 701).

5000 (5) If ABC = -3, add 1 to NAW (the number of variable resettings and Δt cuts).
If NAW $\geq$ 6, write a convergence-failure message and terminate the run.

5001 (6) Test the normal and failure conditions for run termination. If any condition is satisfied, set JCAT = 10 for a subsequent run termination.

7505 (7) Update t and previous values of variables and variable sources for the next Δt.

1253 (8) Compare t with the end of the Δt mode, $T_{end} \equiv$ CT(JCAT, 2):

If $t < T_{end}$, go to the start of the JJ loop.
If $t < T_{end} + 0.999\ \Delta t$, cut Δt so that t coincides with T_{end} and go to the start of the JJ loop.
If $t \geq T_{end} + 0.999\ \Delta t$, start a new Δt mode by increasing JCAT by 1. Examine CT(JCAT, 1) of the new Δt mode:

If CT(JCAT, 1) = 0, go to print out the last results, store the PLOT data on tape, and begin a new run or terminate the run normally.

82 If CT(JCAT,1) < 0, arrange for a new Δt cycle, and modify Δt and t accordingly.

84 If CT(JCAT,1) > 0, modify Δt and t accordingly.

131 Go to the start of the JJ loop.

(9) At t = NB·Δt, store the PLOT data (Ith variable and t values at the JVth point) in VP(I,JV) and

205 TF(JV) arrays.

8766

653 (10) At t = NB·NBB·Δt, print out the source magnitudes and the results (nodal voltages and auxiliary variables). Print out also DELT, NITER, NDELT, and NITRSM.

(11) If the run-time limit is to be exceeded in

1000 20 seconds, print out a message to this effect, and set JCAT = 11 and KPUNCH = 1. If NPLTVT > 0, set KPRPLT = 1.

(12) Test the normal and failure termination conditions and print out which, if any, condition is satisfied.

7002

(13) If CT(JCAT,2) $\leq$ 0, then:

109 (a) If KPLOT = 1 or KPUNCH = 1 or KPRPLT = 1, call Subroutine PLAPR(1) for plotting, print-plotting, and punching cards for a continued run.

(b) Begin a new mode or exit.

85 (14) If CT(JCAT,2) > 0, call PLAPR(-1) and start a new JJ loop.

2. MTRAC Subroutines

Referring to Fig. 32, there are thirty different subroutines in the MTRAC program. These subroutines are listed alphabetically in Table VIII together with their subjects. Also included are the number of words occupied by each subroutine in memory. The main program occupies 8,562 memory words. The memory occupancy for the entire MTRAC program, including the computer-system software, is 45,600 words. All numbers are decimal.

Subroutine AUX( ) allows the user to define auxiliary variables or any other function. This is a source program in the FORTRAN-IV language. This subroutine will be described later in connection with the user's guide.

The READ-WRITE and computation circuit-element subroutines have been described in Sec. II-B, pp. 83-103.

Table VIII

MTRAC SUBROUTINES, THEIR SUBJECTS AND MEMORY OCCUPANCY

SUBROUTINE NAME	SUBJECT	NUMBER OF MEMORY WORDS
AUX( )	Auxiliary variables and functions defined by the user (source program)	*e.g.*, 50
CAP	Capacitors	377
CORE	Magnetic cores	2,481
DIOD	Diodes	586
ELEM	Control of circuit-element subroutines	206
HOLLER(,)	Storing an alphanumeric constant	11
IND	Inductors	583
INTIAL	Solution of initial conditions	254
IVS	Current and floating-voltage sources	218
PAREN(,)	Fixing right-most parenthesis	95
PLAPR( )	Control of plotting subroutines	941
PLT(, , ,)	Plotting	1,958
PRPLT(, , ,)	Print-plotting	655
RES	Resistors	227
RWCAP	READ-WRITE capacitor data	388
RWCND	READ-WRITE continued-run data	509
RWCORE	Control of Subroutines RWCORF and RWCORH	48
RWCORF	READ-WRITE Ø(F) magnetic-core data	2,228
RWCORH	READ-WRITE B(H) magnetic-core data	3,046
RWDIOD	READ-WRITE diode data	439
RWIND	READ-WRITE inductor data	456
RWIVS	READ-WRITE current and floating-voltage source data	369
RWRES	READ-WRITE resistor data	389
RWRUNC	READ-WRITE run control of each mode	1,757
RWTRAN	READ-WRITE transistor data	606
RWZEN	READ-WRITE zener-diode data	376
SECOND( )	Recording run time	20
SEQSOL( )	Solution of matrix equations	115
TRAN	Transistors	1,438
ZEN	Zener diodes	285

Subroutine HOLLER(,) assigns a storage location and a name for an alphanumeric (Hollerith) constant.

Subroutines PLAPR( ), PLT(, , ,) and PRPLT(, , ,) handle plotting and print-plotting of the specified variables. Subroutine PLAPR( ) also handles card punching for continued runs. These subroutines may vary among different computer systems.

Subroutine SECOND( ) records the run time; it is part of the computer system.

Subroutine SEQSOL( ) solves matrix equations by Gaussian elimination to triangularize the coefficient matrix, followed by back substitution.[20] The details of this subroutine are beyond the scope of this report.

The remaining three subroutines are outlined as follows in the order of the call by the main program, *i.e.*, RWRUNC, INTIAL, and RWCND. Refer to the input-data and the glossary of variables in Vol. II.

a. Subroutine RWRUNC (*cf.* Volume II, pp. 101-107)

Statement No.	The run-control specifications are read in and printed out, as follows:
5551	(1) Read in and print out CKT, LIMITT, KONT, and DMAX (Card 00000).*
5550	(2) If end of file, print message to this effect and exit. Set run-time limit, TLIMIT = LIMITT-20 (seconds).
	(3) Read in the MONITOR printout data (Card 00001).
	(4) Initialize NDELT = 0, NITRSM = 0, JCAT = 1, and JKKJ = 0.
	(5) Read in and print out NSVVAR, NSVPHI, and NSVIL (Card 00002). If NSVVAR = 1, set NSVPHI = 1 and NSVIL = 1.
	(6) Read in and print out ITRTST, MAXMIN, KFMAX, RELER, ABSER, and N3B (Card 00003). Set ITRMAX = MAXMIN. If not specified, set RELER = 10^{-6} and ABSER = 10^{-6} volt.

* Card No. in parentheses is an input-data card. See Sec. I, pp. 1-59, in Vol. II.

5517 (7) Initialize to zero TOO(J) for each source, TE and JKLJ. Initialize JV = 1, TIMEIN = TE, and JKLJ = 0.

(8) Test and set the MONITOR printout conditions. Write which subroutines are specified for the MONITOR printout. If NDTMIN $\leq$ NDELT, turn on the conditions to monitor. If NDTMAX < NDELT, turn these conditions off.

1112 (9) Read in and print out NV, NAUX, NG, ND, NB and NBB (Card 01000).
Set NZ = NV + NAUX.

(10) Read in and print out NPLTVT, N1B, and NPLTVV (Card 01010).

(11) If NPLTVT > 0, then read in, process, and print out the PLOT data:
(a) Read in and print out KPLOT, KPRPLT, SCALE, IPX, PRSYMB, and KPFIT (Card 01011).
(b) Read-write names and index numbers of NPLTVT variables to be plotted. If N1B > 0, V *vs.* V will also be plotted.
38 (c) Read-write names and index numbers of NPLTVV pairs of variables to be plotted *vs.* each other.
(d) Check whether all V-*vs.*-V variables are included in the V-*vs.*-*t* list. If not,
2231 print a message and exit.
(e) Compute the total number of plots, K = NPLTVT·(N1B + 1) + NPLTVV. Check whether the PLOT scales are to be determined by the program (SCALE = 0). If K > 0 and SCALE = 1, read in and print out units, scales, and frame boundaries for the *X* and *Y* variables of each plot.

8990 (12) Check whether NV $\leq$ 60, NZ $\leq$ 70, and ND $\leq$ 18. If not, print an appropriate message and exit.

3417 (13) If variable values are not to be saved (NSVVAR = 0), then set to zero Arrays V(J) and V1(J).

7722 (14) Read in and print out the temperature and the Δt values to be used in different periods throughout the transient time.

(15) If NG = 1, read in initial values.
134

(16) If initial conditions are to be solved (NG = -1), set $\Delta t = 10^{20}$.

(17) Read in the variable-source data, including radiation photocurrent and grounded dc voltage sources. For each Jth source (J = 1 for photocurrent; J = 2 for Source No. 1; etc.),
500 store NI and NI2 (see Cards 03000, 03010, 03020, etc.) in the arrays ST(J,1) and ST(J,2), respectively, and the initial slope of a variable source in the Array E2(J). For each Kth source (K = 1 for Source No. 1; K = 2 for Source No. 2; etc.; K = 19 for photocurrent), store the initial source value
4447 in Array EI(K). If DMAX > 0, read in the deviation signs of each source and change the source magnitudes accordingly. Print out the source data.

118 (18) Read in and print out the normal and failure conditions for run termination (Cards 03500 through 03611 etc.).

b. Subroutine INTIAL (*cf.* Volume II, pp. 156-157)

Initial conditions are solved along the following steps:

Statement No.

(1) If NSVVAR = 1, print out a message of inconsistent specifications, and exit.

400 (2) Call Subroutine ELEM, Part (1), to fill the H and T arrays.

8872 (3) Call Subroutine SEQSOL(ERR) to solve $V] = T]/[H]$. If the solution fails, call Subroutine ELEM to fill Array H, print out the H values, and exit.

2001 (4) Test whether the solution of V(J)(for J = 1 to NV) has converged by comparing V(J) with V1(J). If convergence has not been achieved, set JKKJ = 1.

174 (5) Call Subroutine ELEM, Part (2), to compute currents of nonlinear circuit elements (diodes and transistors) and test their convergence.

175 (6) Store V(J) in V1(J) for J = 1 to NV.

(7) If convergence has been achieved (JKKJ $\leq$ 0 and ABC $\leq$ 0), return.
6011 Otherwise, reset JKKJ to zero and add 1 to JKLJ (the number of iterations).

(8) Compare JKLJ with 200:
If JKLJ < 200, start a new iteration.
410 If JKLJ = 200, store V1(J) in V2(J) (for J = 1 to NZ) and start the last iteration.

411 If JKLJ = 201, compute and print out the percent error, 100[V1(J) - V2(J)]/V1(J) (for J = 1 to NZ), and return.

c. Subroutine RWCND (*cf.* Volume II, pp. 152-153)

Read in and print out the following continued-run data.

Statement No.

(1) NDELT, TE, DELT, DTM1, DTM2, and NITRSM.

(2) Variable-source data, TOO(I) for I = 1 to 19, and EP1(I) for I = 1 to ND1.

(3) Nodal and auxiliary unknowns, V1(J) for J = 1 to NZ.

2309 (4) Capacitor currents, CIC(I) for I = 1 to NC.

2313 (5) Inductor currents, CIL(I) for I = 1 to NL.

2316 (6) Diode variables: DV(I), DR(I), and DI(I) for I = 1 to NDIODE.

2319 (7) Transistor variables: BCV(I), BCR(I), BCI(I), BEV(I), BER(I), and BEI(I) for I = 1 to NTRANS.

2318 (8) Magnetic-core variables: FM3(I), FM2(I), FM1(I), PHIM3(I), PHIM2(I), PHIM1(I), PHDTEM1(I),

2326 PHDTM2(I), PHDTM1(I), and, for J = 1 to NW(I), winding currents CICOR1(I,J) for I = 1 to NCORE.

(9) Return.

D. Program Segmentation

The memory occupancies of the main program and the subroutines of MTRAC are listed in Table VIII. The total memory occupancy is 45,600 words (during loading, the required occupancy is 51,100 words). In order to cut this number down, we have divided the MTRAC program into five segments that are listed in Table IX. The maximum memory occupancy (including loading) is thereby reduced to 36,400 words, or 71 percent of of the unsegmented occupancy.

Initially, Segment FIXED is loaded into memory and stays there during the entire run. The remaining segments are loaded into memory following instructions from the main program. See Fig. 32.

Table IX

SEGMENTATION OF MTRAC

SEGMENT NAME	SUBROUTINES INCLUDED	MEMORY OCCUPANCY (words)
FIXED	MTRAC (the main program) including the common storage, AUX( ), HOLLER(,), and SECOND( )	28,456
RWDATA	RWRUNC, RWIVS, RWRES, RWCAP, RWIND, RWZEN, RWDIOD, and RWTRAN	7,912
RWCORS	RWCORE, RWCORF, RWCORH, and RWCND	7,215
SOLVE	SEQSOL, INTIAL, ELEM, IVS, RES, CAP, IND, ZEN, DIOD, TRAN, and CORE	7,785
PLOTS	PLAPR( ), PRPLT(, , ,), PLT(, , ,), and PAREN (,)	6,287

Segment RWDATA is loaded into memory in order to read in and print out the general input data. Using this segment and Segment RWCORS, which is loaded subsequently into memory, the input data of the circuit elements are read in and printed out.

If the run is continued (KONT = 1), then Segment RWCORS is also used to call Subroutine RWCND in order to read in and print out the continued-run data. This is followed by loading Segment SOLVE into memory. If the run is not continued (KONT = 0), then Segment SOLVE is used to compute (partially if NG $\neq$ -1, or fully if NG = -1) the initial conditions.

Segment SOLVE is now used to compute the transient solution. Usually, most of the computer-run time is spent on this solution.

When the mode run terminates or results are to be plotted, Segment PLOTS is loaded into memory in order to print-plot, plot, and punch cards.

If the mode has not terminated, then Segment SOLVE is reloaded to continue the transient solution; otherwise, Segment RWDATA is reloaded to either read in new data or exit.

III APPLICATION

We shall first describe how the auxiliary variables and input-data cards are entered by the user, and then apply MTRAC to transient analyses of several magnetic-core circuits.

A. User's Guide

1. Subroutine AUX(K)

Subroutine AUX(K) is written in the FORTRAN-IV language, and is compiled as a source deck before the MTRAC program is transferred from a binary storage (on disk or on tape) into memory.

As shown in Fig. 31, Subroutine ELEM calls Subroutine AUX(K) twice:

(1) AUX(0) is called in Part (1), before calling the circuit-element subroutines to fill the H and T arrays, and

(2) AUX(1) is called in Part (2), after convergence has been achieved.

Calling AUX(0) in Part (1) allows the user to define variable sources by arbitrary functions, new circuit-element models that will contribute to the H and T arrays, etc. The corresponding cards are inserted between Statement No. 9000 and the following RETURN card (*cf.* the listing of Subroutine AUK(K) shown later in Fig. 36, p. 126).

Calling AUX(1) in Part (2) allows the user to define NAUX auxiliary unknowns that will be part of the printed output. If specified by the user, some or all of these auxiliary unknowns will also be part of the print-plotted and/or plotted output. An auxiliary unknown may be voltage difference, current, F, ϕ, or $\dot{\phi}$ of a given core, power, or any legitimate function of recognized program variables. The NAUX auxiliary unknowns are stored in the V( ) array after the NV nodal voltages have been stored. Hence, the NAUX auxiliary unknowns are defined in Subroutine AUX(K) as V(NV + 1) through V(NV + NAUX). Instructions for defining the most common auxiliary unknowns are given as follows.

Voltage difference: Use unknown nodal voltages. For example, in a circuit with NV = 15 floating nodes, the voltage from Node No. 8 to Node No. 12 may be defined as

$$V(18) = V(8) - V(12) \quad .$$

Current: Use one of the following program names in order to define a current through an Nth circuit element.

Circuit element	Program name for current
Current source or floating-voltage source	CIIVS(N)
Resistor	CIR(N)
Capacitor	CIC(N)
Inductor	CIL(N)
Zener diode	CIZ(N)
Diode	CIDIOD(N)
Transistor	
Collector	CICOLC(N)
Emitter	CIEMTR(N)
Core, Jth winding	CICORE(N,J) .

The references for positive directions of these currents are defined in the instructions for input-data cards, which are given on pp. 1-46 of Vol. II. These directions conform to the corresponding directions in Figs. 7 and 11, 8, 12, 13, 14, 15, 18, and 19. For example, a current through the 3rd winding of the 2nd core, flowing in the negative direction (top terminal to bottom terminal), may be defined as

$$V(22) = -CICORE(2,3) \quad .$$

F, ϕ, and $\dot{\phi}$: Use the following program names.

Variable of Nth core	Program name
F	F(N)
ϕ	PHI(N)
$\dot{\phi}$	PHIDOT(N) .

For example, the flux of Core 1 may be defined as

$$V(27) = PHI(1) \quad .$$

Power: Use the proper formula. For example, the power dissipated in Resistor R5, which is connected between Nodes 3 and 6, may be defined as

$$V(25) = [V(3) - V(6)] * CIR(5) \quad .$$

There are two exceptions: The power dissipated in diodes and transistors are computed by Subroutines DIOD and TRAN, and may be defined by using the following program names.

Element	Program name
*N*th diode	PWCR(N)
*N*th transistor	PWTR(N) .

2. Input Data Cards

Input data are entered in fixed formats within the first 72 columns. Usually, the 0-72 columns of an input-data card are divided into six sections, each consisting of 12 columns. An I12 format is used for an integer number, an E12.5 format for a floating number, an F12.0 format for an integer number that is treated in the program as a floating number, and an A6 (or 2A6) format for an alphanumeric name of up to six (or up to 12) characters.

The input-data cards are designated in Columns 76-80 by 5-digit identification numbers, which are not read in by the program. Although these numbers are not always consecutive (there are many gaps), the cards must be fed in the order of increasing identification numbers. The total number of input-data cards is not fixed, since it depends on the amount of information to be read in. In Fig. 29, for example, no card will follow the first card if NCE = 0, where NCE is the number of the circuit elements of a given type.

The instructions for the order and contents of the input-data cards are given on pp. 1-46 of Vol. II. The examples given there usually have no correlation with each other.

For a worst-case analysis, the maximum deviation DMAX (in percent) of the parameter values is larger than zero. In this case, the program multiplies the value of each input parameter by [1 + (S · DMAX/100)], where S is equal to +1, 0, or -1. The value of S depends on the sign of the parameter change in a direction to usually worsen the circuit behavior. If DMAX > 0, then input-data cards that contain the S values must follow the input-data cards of each circuit-element type. The instructions for these cards are given on pp. 47-59 of Vol. II. Again, the examples given have no correlation with each other. Note that deviation of parameter values other than ±DMAX percent may also be inserted by letting the values of S be different from +1, -1, or 0.

If the run is continued, the cards that had been punched at the end of the previously terminated run are entered following the input-data card deck.

3. Convergence Problems

Not every iterative computation of circuit variables is guaranteed to converge. Usually, convergence is not achieved if the convergence is slow and the values of ITRTST and MAXMIN (Card 00003) are too low. In this case, before having the chance to converge, Δt is cut down very severely and, as a result, terms that include Δt in the denominator blow up due to the computer round-off error.

Convergence may be slowed down by such factors as the following:

(1) An increase in the number of nonlinear circuit elements, *i.e.*, twice the number of transistors plus the number of diodes plus the total number of core windings.

(2) Very slow rate of magnetic flux switching.

(3) Transistor switching from one state to another.

If a run is terminated due to a failure to converge, then the following steps are suggested to aid convergence:

(1) Check the printout for possible errors in the input data.

(2) Increase the values of ITRTST and MAXMIN in Card 00003. (Quite frequently, this step is all that is required.)

(3) Set LASTIC = 0 in Card 80001.

(4) Check whether the maximum allowed relative and absolute errors of core-winding currents in Cards 80001 match the circuit conditions. There are optimum values for these errors which maximize convergence efficiency.

(5) Decrease the values of shunt resistances across circuit elements that limit branch currents to extremely low values.

(6) Obtain MONITOR printout (by setting the controls in Card 00001) and examine the values of the results. If any results make no sense, trace back to find the source of error.

4. Core Parameters

There are twenty-four "ϕF" core parameters, half of which describe the dynamic properties of the core (see Cards 80120 through 80124). Many a user would be unable to provide the values of all of these parameters. In such a case, if blanks are permitted, it is highly recommended to leave these values blank rather than enter estimated values. The estimated values would most likely be worse off than the values computed internally by the program from other given parameters. In addition, computed dynamic parameters would yield a smoother $\dot{\phi}_p$(F) function, thus improving the convergence conditions. However, a minimum of four dynamic parameters (F_0'', F_B, F_0, and ρ_p) must be entered by the user. Fortunately, these parameters are easily determined from an experimental $\dot{\phi}_p$(F) curve (see Fig. 24).

B. Transient Analyses of Magnetic-Core Circuits

Transient analyses of the following magnetic-core circuits were performed under this project by the MTRAC program:

(1) A core driven by a constant-amplitude current source and loaded by R, L, C, and a diode, all in series. (The same circuit was analyzed by a special program in Reports 3 and 4.)

(2) A core-diode-transistor binary counter. (The same circuit was analyzed by a special program in Report 5.)

(3) A core-diode shift register.

(4) A two-phase current driver.

(5) An information-sensing driver.

The results of the first two circuits are very similar to those in Reports 3 through 5. We shall, therefore, describe the analyses of the last three circuits only.

Before we analyze each circuit separately, let us describe how these three circuits are combined into an overall *timer circuit*, which is shown schematically in Fig. 33.

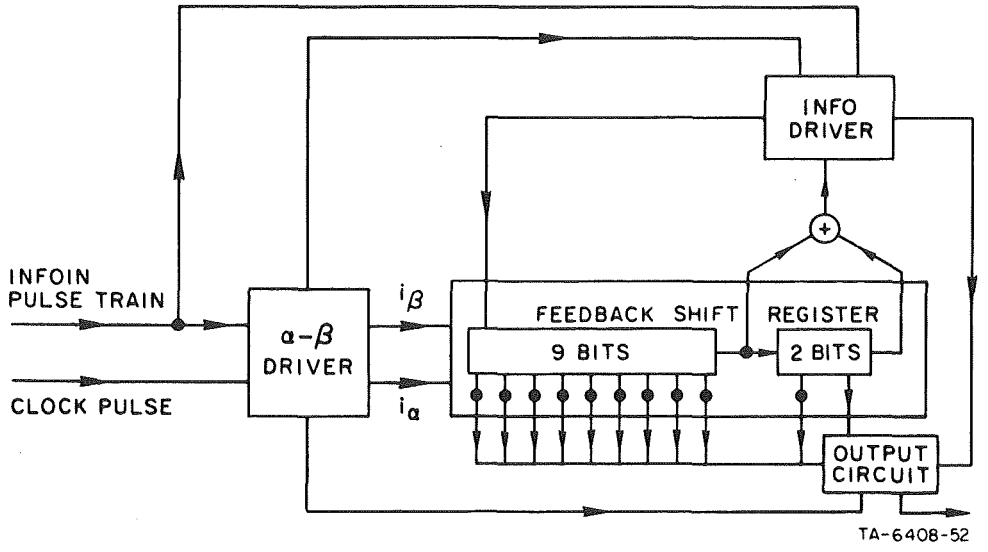


Figure 33 A SCHEMATIC DIAGRAM OF A TIMER CIRCUIT

The *clock pulse* triggers the setting of two cores (Core 1 and Core 2) in the α-β driver, and thus fixes the operation rate of the timer circuit.

The α-β *driver* provides the Advance current pulses i_α and i_β for the feedback shift register and the output circuit, and triggers the INFO driver.

The 9 + 2 = 11-bit core-diode *feedback shift register* has $2^{9+2} - 1 = 2047$ different states. An output signal is generated after the 00000000001 state is reached.

Referring to Fig. 34, there are totally twenty-five cores in the timer circuits: (1) $2 \times 11 = 22$ shift-register cores, Cores α_1 through α_{11} (which are transmitters at the α phase) and Cores β_1 through β_{11} (which are transmitters at the β phase), (2) an input core, Core β_0, which receives a ONE or a ZERO state at the α phase and transmits this

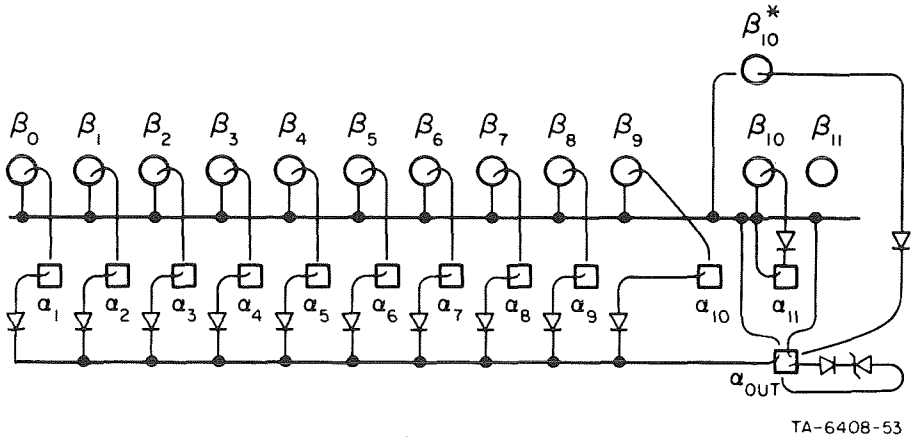


Figure 34 TRANSFER OF STATES IN A TIMER CIRCUIT DURING THE β PHASE (Transmitting β cores are designated by circles; receiving α cores are designated by squares)

state to Core α_1 at the β phase, (3) an auxiliary core, Core β_{10}^*, which receives from Core α_{10} (at the α phase) the same state received by Core β_{10}, and (4) the output core, Core α_{out}, which stays in the ZERO state until Cores β_0 through β_9 transmit ZERO states and Cores β_{10} and β_{10}^* transmit ONE states (at the β phase). The logical function

$$\alpha_{out} = \overline{\beta}_0 \cdot \overline{\beta}_1 \cdot \overline{\beta}_2 \cdot \overline{\beta}_3 \cdot \overline{\beta}_4 \cdot \overline{\beta}_5 \cdot \overline{\beta}_6 \cdot \overline{\beta}_7 \cdot \overline{\beta}_8 \cdot \overline{\beta}_9 \cdot \beta_{10}^*$$

is realized by an inhibit-MMF logic, as is evident from Fig. 34.

The INFO driver is triggered at the α phase by (1) a ONE pulse in the INFOIN pulse train, while the register is loaded initially to the desired state, or (2) the i_α current pulse, when either Core β_9 or Core β_{11} receives a ONE, but not both (an EXCLUSIVE-OR function, characteristic of feedback shift registers). When triggered, the INFO driver generates a current pulse which (1) sets Core β_0 to a ONE state, and (2) drives Core α_{out} toward negative saturation. If at the outset of the α phase the state 00000000001 is stored in Cores α_1 through α_{11}, then Core α_{out} will be cleared from a ONE state to a ZERO state, thus generating an output signal.

Let us now describe the operation and the MTRAC analysis of each circuit separately.

1. A Core-Diode Shift Register

a. Circuit Operation

The circuit diagram of a two-bit core-diode shift register and part of the driver circuit for one of its two phases (the α phase) is shown in Fig. 35. The driver circuit of the β phase is identical with that of the α phase. From the view point of flux-gain behavior, this circuit may simulate a 12-bit shift register containing any information pattern. From the view point of loading the driver circuit, this circuit may represent a 12-bit shift register with all ONEs, all ZEROs, or six ONEs and six ZEROs; this is achieved by letting Winding No. 1 of each transmitter core represent six such windings in series across identically switching cores (see Sec. I-C-1-b, pp. 43-45).

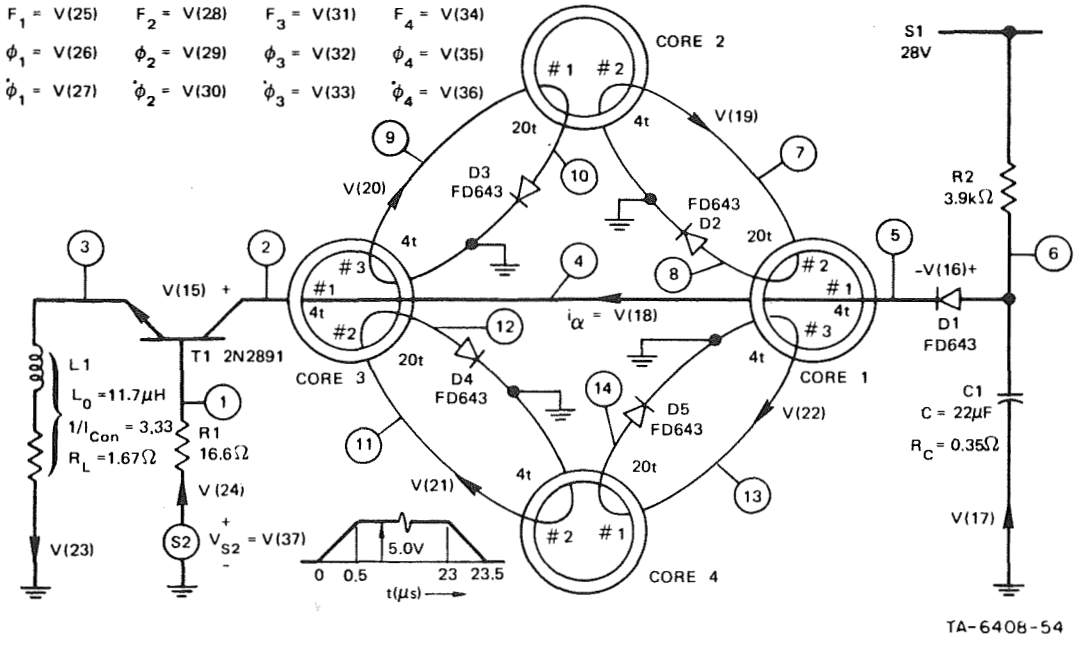


Figure 35 A CORE-DIODE SHIFT-REGISTER CIRCUIT

Consider the α phase in which Cores 1 and 3 are transmitters, while Cores 2 and 4 are receivers. Initially, $\phi_2 = \phi_4 = -\phi_r$, but ϕ_1 and ϕ_3 may have different values representing states of ONE and ZERO.

Also, Capacitor $C1$ is fully charged from the voltage source $S1$ to $V_{S1} = 28.0V$. The second voltage source, V_{S2}, represents the voltage across a zener diode whose nominal voltage is $4.7V$. The zener diode is part of the α-β driver, which is shown later (see Zener-Diode $Z1$ in Fig. 40). Resistor $R1$, Transistor $T1$, and Inductor $L1$ represent two similar circuits in parallel in the actual circuit; the values of each $R1$, R_L, L_0, and $1/I_{con}$ of each of these actual elements are twice those shown in Fig. 35.

As V_{S2} rises to its constant value of 5.0 volts, Transistor $T1$ enters the active region and the drive current, i_α, rises to a constant value and stays at this value as long as V_{S2} is constant. Consequently, Cores 1 and 3 are switched to negative saturation, and the resulting forward loop currents switch Cores 2 and 4 to the initial states of Cores 1 and 3, respectively. Succeeding modes in which the cores interchange functions may be added to the analysis.

b. Input Data

Our objectives are to obtain a computer solution of the initial conditions and a transient analysis during the α phase ($0 \leq t \leq 24.0\ \mu s$), and to plot the resulting waveforms of $i_\alpha(t)$, the forward loop current, $i_{\ell_f}(t)$, and $\dot{\phi}(t)$ of each core. These resulting waveforms will be verified experimentally.

In Fig. 35, we designate the voltage sources $S1$ and $S2$ and the following circuit elements: Resistors $R1$ and $R2$, Capacitor $C1$, Inductor $L1$, Diodes $D1$ through $D5$, Transistor $T1$, and Cores 1 through 4. The index number and number of turns of each core winding are also designated. We designate the floating nodes by Index Numbers 1 through 14; the corresponding unknown voltages are $V(1)$ through $V(14)$. Twenty-three auxiliary unknowns, which are to be part of the output, are designated by $V(15)$ through $V(37)$ in Fig. 35.

The data entry consists of Subroutine AUX(K), where $V(15)$ through $V(37)$ are defined, and the input-data cards. The listing of Subroutine AUX(K) is shown in Fig. 36. The listing of the input-data cards is shown in Fig. 37, and should be compared with the instruction on pp. 1-46 of Vol. II. Note in Columns 25-36 of Cards 80100 and 80300 that initially, both Cores 1 and 3 are assumed in this example to be in a ONE state.

```
      SUBROUTINE  AUX ( K )                                              AUX 0010
C AUX(K) ALLOWS THE USER TO DEFINE NEW FUNCTIONS AND AUXILIARY VARIABLES

                    COMMON  VARIABLES

C  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      IF( K )9000,9000,9001                                              AUX 0020
 9000 CONTINUE                                                           AUX 0030
C     -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
C     DEFINE FUNCTIONS OF NONSTANDARD MODELS, SOURCES, ETC. WHICH AFFECT
C          THE H AND T MATRIX ARRAYS
C     -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
      RETURN                                                             AUX 0040
 9001 CONTINUE                                                           AUX 0050
C     -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
C     DEFINE AUXILIARY VARIABLES FOR OUTPUT
C     -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
      V(15)=V(2)-V(3)                                                    AUX 0060
      V(16)=V(6)-V(5)                                                    AUX 0070
      V(17)=CIC(1)                                                       AUX 0080
      V(18)=-CICORE(1,1)                                                 AUX 0090
      V(19)=CICORE(1,2)                                                  AUX 0100
      V(20)=CICORE(3,3)                                                  AUX 0110
      V(21)=CICORE(3,2)                                                  AUX 0120
      V(22)=CICORE(1,3)                                                  AUX 0130
      V(23)=CIL(1)                                                       AUX 0140
      V(24)=CIR(1)                                                       AUX 0150
      V(25)=F(1)                                                         AUX 0160
      V(26)=PHI(1)                                                       AUX 0170
      V(27)=PHIDOT(1)                                                    AUX 0180
      V(28)=F(2)                                                         AUX 0190
      V(29)=PHI(2)                                                       AUX 0200
      V(30)=PHIDOT(2)                                                    AUX 0210
      V(31)=F(3)                                                         AUX 0220
      V(32)=PHI(3)                                                       AUX 0230
      V(33)=PHIDOT(3)                                                    AUX 0240
      V(34)=F(4)                                                         AUX 0250
      V(35)=PHI(4)                                                       AUX 0260
      V(36)=PHIDOT(4)                                                    AUX 0270
      V(37)=EI(2)                                                        AUX 0280
      RETURN                                                             AUX 0290
C  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      END                                                                AUX 0300
                                                                  TA-6408-46
```

Figure 36 LISTING OF SUBROUTINE AUX(K) FOR THE CORE-DIODE SHIFT-REGISTER CIRCUIT IN FIG. 35

```
0000000001111111111222222222233333333334444444445555555555666666666677777777778
1234567890123456789012345678901234567890123456789012345678901234567890123456789
CKT 8F                180                                                      0000
0,0,0,0,0,0,0,0,0,0,0,                          10                               0001
                                                                                 0002
          14            40           3            0           0                  0003
          14            23-          1            2           1            1     01000
           4                                                                     1010
           1             1           1          350           *            2     1011
IALPHA    18ILFWD      19PHDOT1     27PHDOT2     30                              01020
TIME        MICROSECONDS 2.54     -06-1.0     -06 12.0     -06 PLOT 1           01040
IALPHA      AMPERE       1.27       -0.5         5.0         PLOT 1               01041
TIME        MICROSECONDS 2.54     -06-1.0     -06 12.0     -06 PLOT 2           01042
ILFWD       AMPERE       0.127      -0.05        0.5         PLOT 2               01043
TIME        MICROSECONDS 2.54     -06-1.0     -06 12.0     -06 PLOT 3           01044
PHDOT1      VOLT/TURN    0.127      -0.5         0.05        PLOT 3               01045
TIME        MICROSECONDS 2.54     -06-1.0     -06 12.0     -06 PLOT 4           01046
PHDOT2      VOLT/TURN    0.635      -0.25        2.5         PLOT 4               01047
 298.0                     4                                                     01100
 1.0     -08 2.5      -06 1.0     -07 12.0     -06 1.0     -06 23.0     -06     01200
 1.0     -07 24.0     -06                                                       01201
                                                                                 03000
          1-            1                                                        03010
 28.0                                                                            03011
           4                                                                     03020
 0.0         0.0          5.0          0.5     -06 5.0          23.0     -06     03021
 0.0         23.5     -06                                                       03022
                                                                                 03500
                                                                                 03600
                                                                                 10000
           2                                                                     20000
S          2             1 16.6                                                 20010
S          1             6 3.9       +03                                        20020
           1                                                                     30000
G          0             6 22.0      -06 0.35                                   30010
           1                                                                     40000
           3G            0 11.70     -06 1.6735                    3.33         40010
                                                                                 40011
                                                                                 50000
           5                                                                     60000
           6             5                                                       60010
 9.7465  -09 1.96          6.05      +09 1.0     -12 0.8          1.0      -09     60011
           8G            0            1                                        60020
          10G            0            1                                        60030
          12G            0            1                                        60040
          14G            0            1                                        60050
           1                                                                     70000
                         1            2            3                           70010
 64.0        3.0           5.30      -09 1.00    -06 0.1      -09 1.2           70011
 20.0    -12 0.8           10.0      +06 0.1     -09 1.0          40.0     -12     70012
 0.8         10.0      +06                                                       70013
           4                                                                     80000
 1.0     -04 1.0      -04                                          10           1  80001
           3                1.0                                                  80100
           4             5            4 0.260                     6              80101
           7             8           20 0.81                                     80102
G          0            13            4 0.225                                   80103
 0.147       0.09                        35.0    E-08                            80120
 0.21                                              -13.0    E-08                80121
 147.1       19.81          15.23                                               80122
                                       0.2863                                    80123
 1.7897      0.5795         1.50                                                80124
           2             1-1.0                                                 80200
           9            10           20 1.12                                   80201
G          0             7            4 0.165                                   80202
           3             1 1.0                                                 80300
           2             4            4 0.015                     6             80301
          11            12           20 0.81                                   80302
G          0             9            4 0.225                                   80303
           2             1-1.0                                                 80400
          13            14           20 1.12                                   80401
G          0            11            4 0.165                                   80402
                                                                          TB-6408-47
```

Figure 37 LISTING OF THE INPUT-DATA CARDS FOR THE CORE-DIODE SHIFT-REGISTER CIRCUIT IN FIG. 35

c. Results

The computer printout of the input data, some of the computed parameters, and the beginning of the computed results are shown in Fig. 38.

The resulting computed plots are compared with experimental oscillograms in Fig. 39. Referring to Fig. 35, these include the waveforms of (a) $i_\alpha \equiv$ IALPHA = $V(18)$, (b) $i_{\ell_f} \equiv$ ILFWD = $V(19)$, (c) $\dot{\phi}_1 \equiv$ PHDOT1 = $V(27)$, and (d) $\dot{\phi}_2 \equiv$ PHDOT2 = $V(30)$. The experimental and computed waveforms of $\dot{\phi}_2(t)$ are very close to each other.

2. A Two-Phase Current Driver

a. Circuit Operation

The circuit diagram of the two-phase (α-β) current driver is shown in Fig. 40. This current generates the voltage-source pulse V_{S2} in Fig. 35 across Zener-Diode $Z1$ for i_α and a similar voltage pulse across Zener-Diode $Z2$ for i_β. Two modes of operation are distinguished:

Mode I: Cores 1 and 2 are set from $\phi = -\phi_r$ to $\phi \approx \phi_r$. During this mode, Transistors $T1$ and $T2$ are cut off. In order to shorten the computer-run time, the equivalent circuit in Mode I is simplified by eliminating the circuit portion which is associated with Nodes 7 through 15. However, a voltage source, V_{S2}, is added from ground to the top terminal of Winding No. 1 of Core 1.

Mode II. The current through Winding No. 1 of Core 1 is essentially cut off, and the current stored in Inductor $L1$ unblocks Diode $D1$. Consequently, a negative $\dot{\phi}_1$ is generated and Transistor $T1$ is turned on; Core 1 is then cleared to $\phi = -\phi_r$ by a blocking-oscillator action as the current through Inductor $L2$ builds up. With Node 9 being tied to the base of an output transistor (Transistor $T1$ in Fig. 35), the latter is in the active region while Core 1 is switching. When Core 1 completes switching, Transistor $T1$ is cut off, and the current stored in Inductor $L2$ unblocks Diode $D2$. Consequently, a negative $\dot{\phi}_2$ is generated and Transistor $T2$ is turned on; Core 2 is then cleared to $\phi = -\phi_r$ by a blocking-oscillator action. With Node 14 tied to the base of a second output transistor (not shown in either Fig. 35 or Fig. 40), the latter is in the active region until Core 2 completes its switching.

C I R C U I T CKT AF

```
OVERALL-RUN DATA
LIMITT=180 (COMPUTER-RUN PROCESS TIME = LIMITT-20  SECONDS)
KONT = -0 (1 = CONTINUED RUN, 0 = OTHERWISE)
SAVE-VARIABLE DATA
NSVVAR  =  -0 (1 = SAVE ALL TIME VARIABLES FOR THE FOLLOWING MODE, 0 = OTHERWISE)
NSVPHI  =  -0 (1 = SAVE ALL FLUX VALUES FOR THE FOLLOWING MODE, 0 = OTHERWISE)
NSVIL   =  -0 (1 = SAVE INDUCTOR-CURRENT VALUES FOR THE FOLLOWING MODE, 0 = OTHERWISE)

CONVERGENCE DATA
ITRTST  = 14 (NUMBER OF TEST ITERATIONS BELOW WHICH DELT IS MULTIPLIED BY 2 AND ABOVE WHICH DELT IS DIVIDED BY 2)
MAXMIN  = 40 (INITIAL MAXIMUM NUMBER OF ITERATIONS BEFORE VARIABLE VALUES ARE RESET AND DELT IS DIVIDED BY 4)
KFMAX   =  3 (MAXIMUM NUMBER OF TRIALS BEFORE ITRMAX IS CHANGED)
RELER= 1.00000E-06 ABSER =  1.00000E-06 FOR INITIAL CONDITIONS

GENERAL DATA
NV  = 14   (NUMBER OF UNKNOWN NODAL VOLTAGES)
NAUX= 23   (NUMBER OF AUXILIARY UNKNOWNS)
NG =  -1   (-1=CALCULATE VALUES, 0=ZERO VALUES, 1=READ VALUES)
N3R=  -0   (-1 = INITIAL CONDITIONS ONLY, 0 = HALT IF ERROR IN I.C., 1 = PRINT IF ERROR IN I.C. BUT PROCEED)
ND =   2   (NUMBER OF VARIABLE SOURCES)
NB =   1   (RATIO OF COMPUTED POINTS TO PLOTTED POINTS)
NBR=   1   (RATIO OF PLOTTED POINTS TO PRINTED POINTS)

PLOT AND PRINT-PLOT DATA
SCALE  =  1  (1 = READ SCALE FROM CARDS, 0 = OTHERWISE)
NPLTVT =  4  (NUMBER OF VARIABLES TO BE PLOTTED)
NPLTVV = -0  (NUMBER OF V-VS.-V PLOTS)
N1B    = -0  (1 = PLOT VDOT VS. V PHASE PLANES, 0 = OTHERWISE)
KPRPLT =  1  (1 = PRINT PLOT, 0 = DO NOT PRINT PLOT AT NORMAL EXIT)
KPLOT  =  1  (1 = PLOT, 0 = DO NOT PLOT)
IPX    =350  (NUMBER OF PLOTTED POINTS PER FRAME)
KPFIT  =  2  (0=INDEPENDENT SCALES AND BOUNDARIES FOR FRAMES, 1=SAVE Y SCALES AND Y BOUNDARIES,
              2=SAVE X SCALES, Y SCALES, AND Y BOUNDARIES, 3=SAVE ALL SCALES AND BOUNDARIES)

PLOTS OF VARIABLE VS. TIME
(SAVED BUT NOT PLOTTED IF VARIABLE NO. IS NEGATIVE)
 VARIABLE
NAME    NO.
IALPHA  18
ILFWD   19
PHDOT1  27
PHDOT2  30
```

(a) Page 1

P L O T D A T A

PLOT NO.	X	X-UNITS	X-SCALE	LEFT	RIGHT	Y	Y-UNITS	Y-SCALE	BOTTOM	TOP
1	TIME	MICROSECONDS	2.540E-06	-1.00E-06	1.20E-05	IALPHA	AMPERE	1.270E+00	-5.00E-01	5.00E+00
2	TIME	MICROSECONDS	2.540E-06	-1.00E-06	1.20E-05	ILFWD	AMPERE	1.270E-01	-5.00E-02	5.00E-01
3	TIME	MICROSECONDS	2.540E-06	-1.00E-06	1.20E-05	PHDOT1	VOLT/TURN	1.270E-01	-5.00E-01	5.00E-02
4	TIME	MICROSECONDS	2.540E-06	-1.00E-06	1.20E-05	PHDOT2	VOLT/TURN	6.350E-01	-2.50E-01	2.50E+00

```
ABSOLUTE TEMPERATURE=   2.98000E+02

NO.    DELT       END TIME
 1  1.0000E-08  2.5000E-06
 2  1.0000E-07  1.2000E-05
 3  1.0000E-06  2.3000E-05
 4  1.0000E-07  2.4000E-05
```

(b) Page 2

```
DC SOURCE NO.  1
MAGNITUDE =  2.8000E+01

VARIABLE SOURCE NO.  2
LAST MAGNITUDE HELD
MAGNITUDE       TIME
 0.             0.
 5.0000E+00     5.0000E-07
 5.0000E+00     2.3000E-05
 0.             2.3500E-05
```

(c) Page 3

Figure 38 COMPUTER PRINTOUT FOR THE CORE-DIODE SHIFT-REGISTER CIRCUIT IN FIG. 35

```
RESISTORS
PART
 NO   NODE A   NODE B              RESISTANCE

  1   S  2        1                1.66000E+01

  2   S  1        6                3.90000E+03
```

(d) Page 4

```
 CAPACITORS
PART                                                    SERIES          SHUNT
 NO   NODE A   NODE B              CAPACITANCE        RESISTANCE      RESISTANCE      CURRENT

  1   GROUND      6                2.20000E-05     3.50000E-01     1.00000E+09   -0.
```

(e) Page 5

```
 INDUCTORS
PART                                                    SERIES          SHUNT
 NO   NODE A   NODE B              INDUCTANCE         RESISTANCE      RESISTANCE      INVICON         CURRENT

  1      3     GROUND             1.17000E-05     1.67350E+00     1.00000E+09   3.33000E+00   -0.
```

(f) Page 6

```
DIODES
NO   NODE P   NODE N      ISO          MD          BLO          COD          VPOTD          T D          IPP

 1      6        5     9.747E-09   1.960E+00   6.050E+09   1.000E-12   8.000E-01   1.000E-09   -0.

 2      8     GROUND   9.747E-09   1.960E+00   6.050E+09   1.000E-12   8.000E-01   1.000E-09   -0.

 3     10     GROUND   9.747E-09   1.960E+00   6.050E+09   1.000E-12   8.000E-01   1.000E-09   -0.

 4     12     GROUND   9.747E-09   1.960E+00   6.050E+09   1.000E-12   8.000E-01   1.000E-09   -0.

 5     14     GROUND   9.747E-09   1.960E+00   6.050E+09   1.000E-12   8.000E-01   1.000E-09   -0.
```

(g) Page 7

```
NPN TRANSISTOR
NO=  1

NODE B   NODE C   NODE E
   1        2        3

  BETAN 6.4000E+01   BETAI 3.0000E+00     TE  5.3000E-09     TC  1.0000E-06
  ISC 1.0000E-10     MC 1.2000E+00        COC 2.0000E-11   VPOTC 8.0000E-01     RLC 1.0000E+07
  ISE 1.0000E-10     ME 1.0000E+00        COE 4.0000E-11   VPOTE 8.0000E-01     RLE 1.0000E+07
```

(h) Page 8

```
MAGNETIC CORES


ELASTIC PHIDOT INCLUDED

DO NOT EXIT WHEN FLUX SWITCHING STOPS

RELERR =  1.000E-04          ABSERR =  1.000E-04


PSTEP   =-0.
```

(i) Page 9

Figure 38 Continued

CORE NO. 1

WINDINGS

WINDING NO.	NODE B	NODE T	TURNS	RESISTANCE	INITIAL CURRENT
1	4	5	4	2.60000E-01	-0.
REPRESENTING 6 IDENTICALLY SWITCHING CORES IN SERIES					
2	7	8	20	8.10000E-01	-0.
3	GROUND	13	4	2.25000E-01	-0.

CORE PARAMETERS

OD= 1.47000E-01	ID= 9.00000E-02	TYPE-R	MATERIAL- FERRITE	
PHIS= 3.85000E-07	PHIR= 3.50000E-07	FD1= 1.99500E-01	FDZ= 2.04750E-01	FD2= 2.10000E-01
FD3=-0.	PHID2=-1.30000E-07	HA= 1.47100E+02	HQ= 1.98100E+01	HN= 1.52300E+01
LAMDAD= 3.36058E+01	NUD= 3.44941E+00	FDB= 3.29245E-01	FOPP= 2.86300E-01	LAMDA= 1.09389E+00
NU= 1.24227E+00	FB= 1.78970E+00	F0= 5.79500E-01	ROP= 1.50000E+00	
FB1= 1.00000E+30	F01=-0.	ROP1=-0.		

COMPUTED AUXILIARY PARAMETERS

P1= 5.23115E-08	P2= 1.72549E+00	P3= 1.05643E+00	P4= 3.50000E-08	P5=-3.43978E-07
P6= 2.03788E-05	P7= 1.00000E+00	P8= 0.	P9= 0.	P10= 0.
P11= 0.	P12= 1.05920E-05	P13= 3.19076E-06	P14= 2.32524E-01	P15= 1.09377E-01
P16= 1.85952E-06	P17= 1.78649E-01	P18=-6.06031E-07	P19= 3.48996E-07	P20= 2.21031E-07
ZV = 1.55510E-04				

INITIAL FLUX = 3.50000E-07

(j) Page 10

CORE NO. 2

WINDINGS

WINDING NO.	NODE B	NODE T	TURNS	RESISTANCE	INITIAL CURRENT
1	9	10	20	1.12000E+00	-0.
2	GROUND	7	4	1.65000E-01	-0.

CORE PARAMETERS

OD= 1.47000E-01	ID= 9.00000E-02	TYPE-R	MATERIAL- FERRITE	
PHIS= 3.85000E-07	PHIR= 3.50000E-07	FD1= 1.99500E-01	FDZ= 2.04750E-01	FD2= 2.10000E-01
FD3=-0.	PHID2=-1.30000E-07	HA= 1.47100E+02	HQ= 1.98100E+01	HN= 1.52300E+01
LAMDAD= 3.36058E+01	NUD= 3.44941E+00	FDB= 3.29245E-01	FOPP= 2.86300E-01	LAMDA= 1.09389E+00
NU= 1.24227E+00	FB= 1.78970E+00	F0= 5.79500E-01	ROP= 1.50000E+00	
FB1= 1.00000E+30	F01=-0.	ROP1=-0.		

COMPUTED AUXILIARY PARAMETERS

P1= 5.23115E-08	P2= 1.72549E+00	P3= 1.05643E+00	P4= 3.50000E-08	P5=-3.43978E-07
P6= 2.03788E-05	P7= 1.00000E+00	P8= 0.	P9= 0.	P10= 0.
P11= 0.	P12= 1.05920E-05	P13= 3.19076E-06	P14= 2.32524E-01	P15= 1.09377E-01
P16= 1.85952E-06	P17= 1.78649E-01	P18=-6.06031E-07	P19= 3.48996E-07	P20= 2.21031E-07
ZV = 1.55510E-04				

INITIAL FLUX = -3.50000E-07

(k) Page 11

Figure 38 Continued

CORE NO. 3

WINDINGS

WINDING NO.	NODE B	NODE T	TURNS	RESISTANCE	INITIAL CURRENT
1	2	4	4	1.50000E-02	-0.
REPRESENTING 6 IDENTICALLY SWITCHING CORES IN SERIES					
2	11	12	20	8.10000E-01	-0.
3	GROUND	9	4	2.25000E-01	-0.

CORE PARAMETERS

OD= 1.47000E-01	ID= 9.00000E-02	TYPE-B	MATERIAL- FERRITE	
PHIS= 3.85000E-07	PHIR= 3.50000E-07	FD1= 1.99500E-01	FD2= 2.04750E-01	FD2= 2.10000E-01
FD3=-0.	PHID2=-1.30000E-07	HA= 1.47100E+02	HO= 1.98100E+01	HN= 1.52300E+01
LAMDAD= 3.36058E+01	NUD= 3.44941E+00	FDB= 3.29245E-01	FOPP= 2.86300E-01	LAMDA= 1.09389E+00
NU= 1.24227E+00	FB= 1.78970E+00	F0= 5.79500E-01	ROP= 1.50000E+00	
FB1= 1.00000E+30	F01=-0.	ROP1=-0.		

COMPUTED AUXILIARY PARAMETERS

P1= 5.23115E-08	P2= 1.72549E+00	P3= 1.05643E+00	P4= 3.50000E-08	P5=-3.43978E-07
P6= 2.03788E-05	P7= 1.00000E+00	P8= 0.	P9= 0.	P10= 0.
P11= 0.	P12= 1.05920E-05	P13= 3.19076E-06	P14= 2.32524E-01	P15= 1.09377E-01
P16= 1.85952E-06	P17= 1.78649E-01	P18=-6.06031E-07	P19= 3.48996E-07	P20= 2.21031E-07
ZV = 1.55510E-04				

INITIAL FLUX = 3.50000E-07

(l) Page 12

CORE NO. 4

WINDINGS

WINDING NO.	NODE B	NODE T	TURNS	RESISTANCE	INITIAL CURRENT
1	13	14	20	1.12000E+00	-0.
2	GROUND	11	4	1.65000E-01	-0.

CORE PARAMETERS

OD= 1.47000E-01	ID= 9.00000E-02	TYPE-B	MATERIAL- FERRITE	
PHIS= 3.85000E-07	PHIR= 3.50000E-07	FD1= 1.99500E-01	FD2= 2.04750E-01	FD2= 2.10000E-01
FD3=-0.	PHID2=-1.30000E-07	HA= 1.47100E+02	HO= 1.98100E+01	HN= 1.52300E+01
LAMDAD= 3.36058E+01	NUD= 3.44941E+00	FDB= 3.29245E-01	FOPP= 2.86300E-01	LAMDA= 1.09389E+00
NU= 1.24227E+00	FB= 1.78970E+00	F0= 5.79500E-01	ROP= 1.50000E+00	
FB1= 1.00000E+30	F01=-0.	ROP1=-0.		

COMPUTED AUXILIARY PARAMETERS

P1= 5.23115E-08	P2= 1.72549E+00	P3= 1.05643E+00	P4= 3.50000E-08	P5=-3.43978E-07
P6= 2.03788E-05	P7= 1.00000E+00	P8= 0.	P9= 0.	P10= 0.
P11= 0.	P12= 1.05920E-05	P13= 3.19076E-06	P14= 2.32524E-01	P15= 1.09377E-01
P16= 1.85952E-06	P17= 1.78649E-01	P18=-6.06031E-07	P19= 3.48996E-07	P20= 2.21031E-07
ZV = 1.55510E-04				

INITIAL FLUX = -3.50000E-07

(m) Page 13

Figure 38 Continued

```
    3 INITIAL CONDITION ITERATIONS
ABC=   -1

INITIAL VALUES
SOURCE
  1  2.80000E+01    2  0.

UNKNOWNS
  1  4.59890E-05    2  2.77041E+01    3  1.33508E-10    4  2.77041E+01    5  2.77041E+01    6  2.79890E+01
  7  0.             8  0.             9  0.            10  0.            11  0.            12  0.
 13  0.            14  0.            15  2.77041E+01   16  2.84903E-01   17 -5.59780E-08   18  1.66230E-05
 19  0.            20  0.            21  0.            22  0.            23  7.97779E-11   24 -2.77042E-06
 25 -6.64921E-05   26  3.50002E-07   27  0.            28  0.            29 -3.50000E-07   30  0.
 31 -6.64921E-05   32  3.50002E-07   33  0.            34  0.            35 -3.50000E-07   36  0.
 37  0.

IONIZING PULSE SCALE FACTOR =   0.


DELTA TIME= 1.00000E-08     START TIME=  0.              END TIME= 2.50000E-06

PRO. TIME =     1.312E+01
```

(n) Page 14

```
SOURCE
  1  2.80000E+01    2  1.00000E-01

TIME= 1.00000E-08
UNKNOWNS
  1  9.92559E-02    2  2.77755E+01    3  8.96403E-02    4  2.77703E+01    5  2.77652E+01    6  2.79890E+01
  7  3.20035E-05    8 -4.26539E-03    9 -8.59230E-04   10 -6.99156E-04   11  3.20035E-05   12 -4.26539E-03
 13 -8.59230E-04   14 -6.99156E-04   15  2.76859E+01   16  2.23821E-01   17 -9.36781E-06   18 -6.54216E-06
 19 -4.27448E-07   20 -7.00645E-08   21 -4.27448E-07   22 -7.00645E-08   23  3.82806E-05   24  4.48228E-05
 25  1.73395E-05   26  3.50000E-07   27  2.14812E-04   28 -3.11108E-06   29 -3.50000E-07   30 -7.99835E-06
 31  1.73395E-05   32  3.50000E-07   33  2.14812E-04   34 -3.11108E-06   35 -3.50000E-07   36 -7.99835E-06
 37  1.00000E-01

DELT =  1.00000E-08                  6 ITERATION(S)              NDELT =     1              ITER. SUM =     6

SOURCE
  1  2.80000E+01    2  2.00000E-01

TIME= 2.00000E-08
UNKNOWNS
  1  1.97510E-01    2  2.78416E+01    3  1.53340E-01    4  2.78403E+01    5  2.78390E+01    6  2.79890E+01
  7 -4.95508E-05    8 -1.11653E-03    9 -2.12267E-04   10 -4.59780E-04   11 -4.95508E-05   12 -1.11653E-03
 13 -2.12267E-04   14 -4.59780E-04   15  2.76882E+01   16  1.50026E-01   17 -1.08512E-05   18 -8.00587E-06
 19  3.13874E-07   20  2.27553E-08   21  3.13874E-07   22  2.27553E-08   23  1.42003E-04   24  1.50028E-04
 25  3.83920E-05   26  3.50001E-07   27  5.30542E-05   28  1.71060E-06   29 -3.50000E-07   30  1.23183E-05
 31  3.83920E-05   32  3.50001E-07   33  5.30542E-05   34  1.71060E-06   35 -3.50000E-07   36  1.23183E-05
 37  2.00000E-01

DELT =  1.00000E-08                  7 ITERATION(S)              NDELT =     2              ITER. SUM =    13

SOURCE
  1  2.80000E+01    2  3.00000E-01

TIME= 3.00000E-08
UNKNOWNS
  1  2.95131E-01    2  2.79089E+01    3  1.84071E-01    4  2.79093E+01    5  2.79097E+01    6  2.79890E+01
  7  1.00142E-06    8  3.19783E-04    9  6.50112E-05   10  7.01511E-05   11  1.00142E-06   12  3.19783E-04
 13  6.50112E-05   14  7.01511E-05   15  2.77249E+01   16  7.93176E-02   17 -1.01990E-05   18 -7.35109E-06
 19  1.43617E-07   20  5.24100E-08   21  1.43617E-07   22  5.24100E-08   23  2.85957E-04   24  2.93330E-04
 25  3.24863E-05   26  3.50001E-07   27 -1.62680E-05   28  1.62267E-06   29 -3.50000E-07   30 -3.20897E-07
 31  3.24863E-05   32  3.50001E-07   33 -1.62680E-05   34  1.62267E-06   35 -3.50000E-07   36 -3.20897E-07
 37  3.00000E-01

DELT =  1.00000E-08                  5 ITERATION(S)              NDELT =     3              ITER. SUM =    18

SOURCE
  1  2.80000E+01    2  4.00000E-01

TIME= 4.00000E-08
UNKNOWNS
  1  3.92522E-01    2  2.79765E+01    3  1.85572E-01    4  2.79761E+01    5  2.79756E+01    6  2.79890E+01
  7 -2.42857E-05    8 -4.12934E-04    9 -7.77381E-05   10 -1.99195E-04   11 -2.42857E-05   12 -4.12934E-04
 13 -7.77381E-05   14 -1.99195E-04   15  2.77910E+01   16  1.33890E-02   17 -9.76860E-06   18 -6.94299E-06
 19 -7.33812E-08   20 -2.69796E-08   21 -7.33812E-08   22 -2.69796E-08   23  4.43551E-04   24  4.50494E-04
 25  2.61964E-05   26  3.50001E-07   27  1.94361E-05   28 -8.33117E-07   29 -3.50000E-07   30  6.07459E-06
 31  2.61964E-05   32  3.50001E-07   33  1.94361E-05   34 -8.33117E-07   35 -3.50000E-07   36  6.07459E-06
 37  4.00000E-01

DELT =  1.00000E-08                  4 ITERATION(S)              NDELT =     4              ITER. SUM =    22
```

(o) Page 15

Figure 38 Concluded

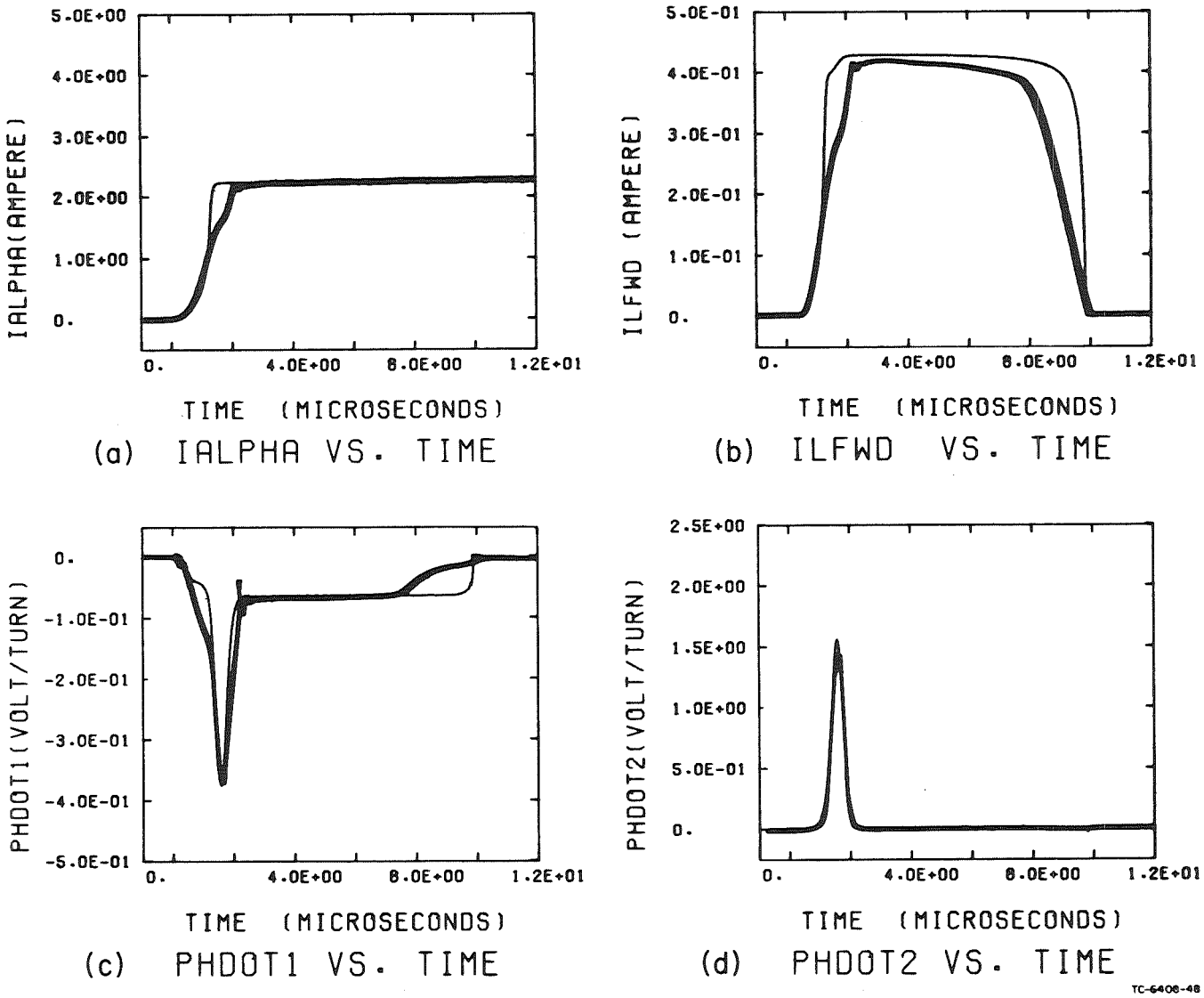


Figure 39 EXPERIMENTAL (heavy line) AND COMPUTED (light line) WAVEFORMS OF i_α, i_F, $\dot{\phi}_1$, AND $\dot{\phi}_2$ OF THE CORE-DIODE SHIFT-REGISTER CIRCUIT IN FIG. 35 FOR THE ALL-ONE CASE AT 25 °C

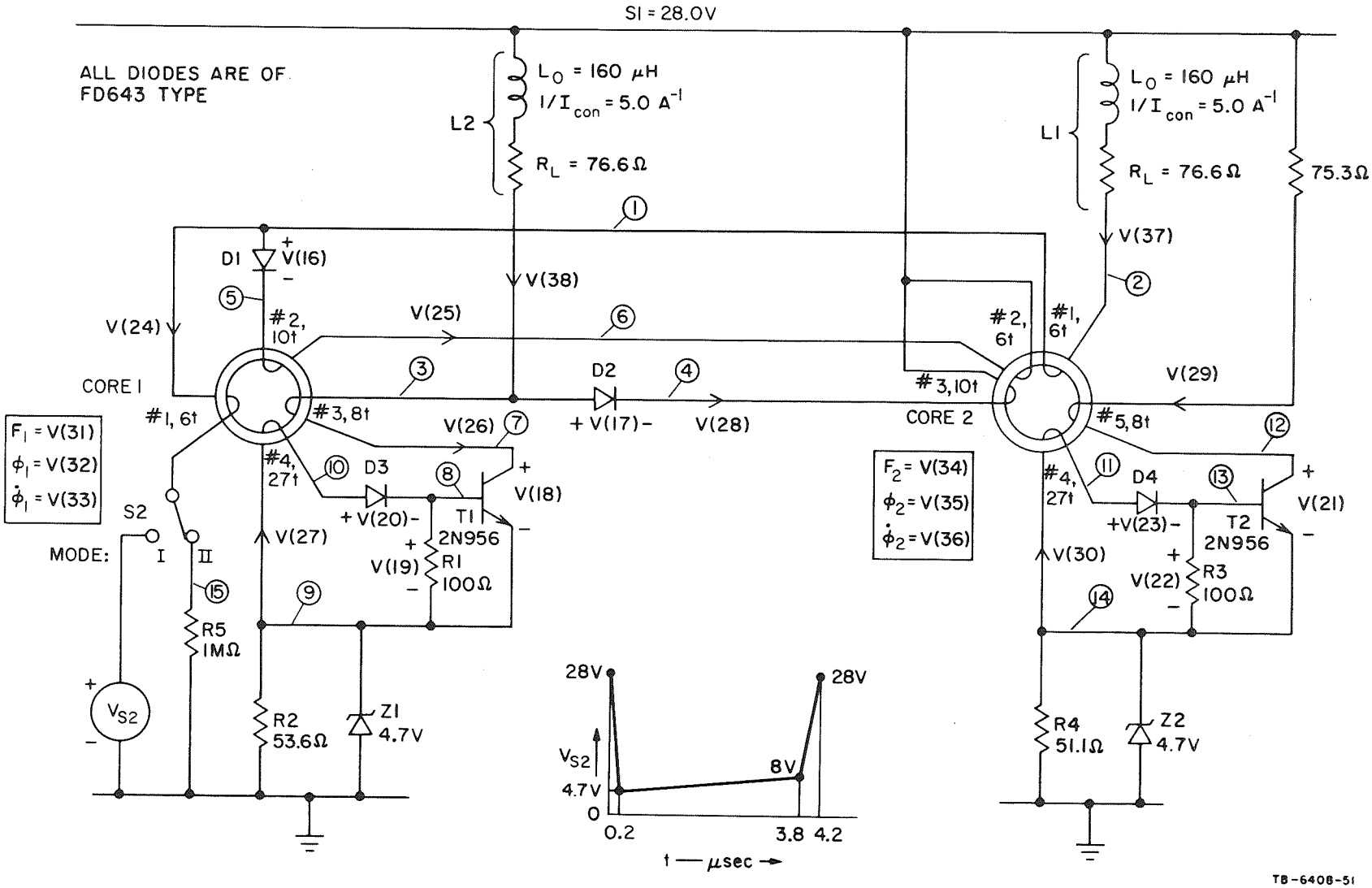


Figure 40 A TWO-PHASE CURRENT-DRIVER CIRCUIT

b. Input Data

In Fig. 40, we designate the voltage sources $S1$ and $S2$ and the following circuit elements: Resistors $R1$ through $R5$, Nonlinear Inductors $L1$ and $L2$, Zener diodes $Z1$ and $Z2$, Diodes $D1$ through $D4$, Transistor $T1$ and $T2$, and Cores 1 and 2. The index number and number of turns of each core winding are also designated. Next, we designate the floating nodes by Index Numbers 1 through 15; the corresponding unknown voltages are $V(1)$ through $V(15)$. Twenty-three auxiliary unknowns, to be part of the output, are marked by $V(16)$ through $V(38)$ in Fig. 40.

The data are entered in a way similar to that described above in connection with the shift-register circuit, Fig. 35.

c. Results

Computed and experimental waveforms of six variables are compared in Fig. 41: i_{L1}, i_{D1}, $\dot{\phi}_1$, and $\dot{\phi}_2$ in Mode *I*, and i_{D3} and i_{cT1} in Mode II.

3. An Information-Sensing Driver

a. Circuit Operation

The circuit diagram of the information-sensing driver is shown in Fig. 42. Compared with Fig. 34, Core 1 corresponds to Core α_{11}, Core 2 to Core β_{11}, Core 3 to Core α_9, Core 4 to Core β_9, Core 5 to Core β_0, Core 6 to Core α_{out}, and Core 7 to Core β^*_{10}.

Referring to Fig. 34, an output signal should be generated at the α phase in which Cores α_1 through α_{11} store the state 00000000001 initially. At this state, Core α_{out} is in a ONE state. Thus, the initial fluxes in Fig. 42 when output signal is to be generated are $\phi_1 = \phi_6 = \phi_r$ and $\phi_2 = \phi_3 = \phi_4 = \phi_5 = \phi_7 = -\phi_r$.

The voltage source V_{S2} represents $V(9)$ in Fig. 40 (the voltage across Zener-Diode $Z1$), and the current source i_{S3} represents the advance current i_a in Fig. 35. Core 1 is cleared by i_{S3}, and the resulting forward loop current sets Core 2. At the same time, Core 3 is driven by i_{S3} further into negative saturation and back and the resulting small forward loop current drives Core 4 in the positive direction by a small amount. Since $\dot{\phi}_2 >> \dot{\phi}_4$ ($\dot{\phi}_2$ is primarily inelastic while $\dot{\phi}_3$ is primarily elastic), the

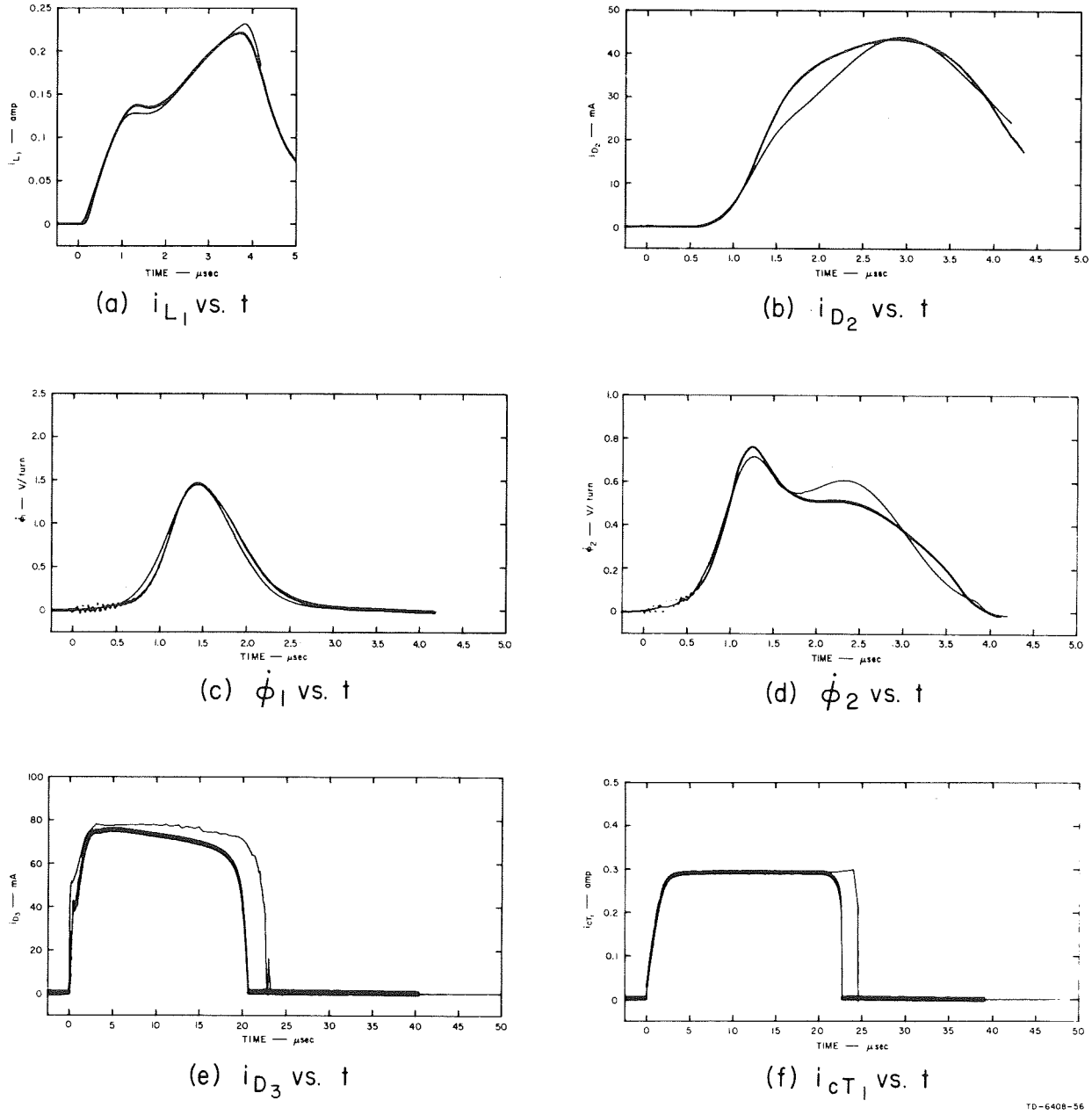
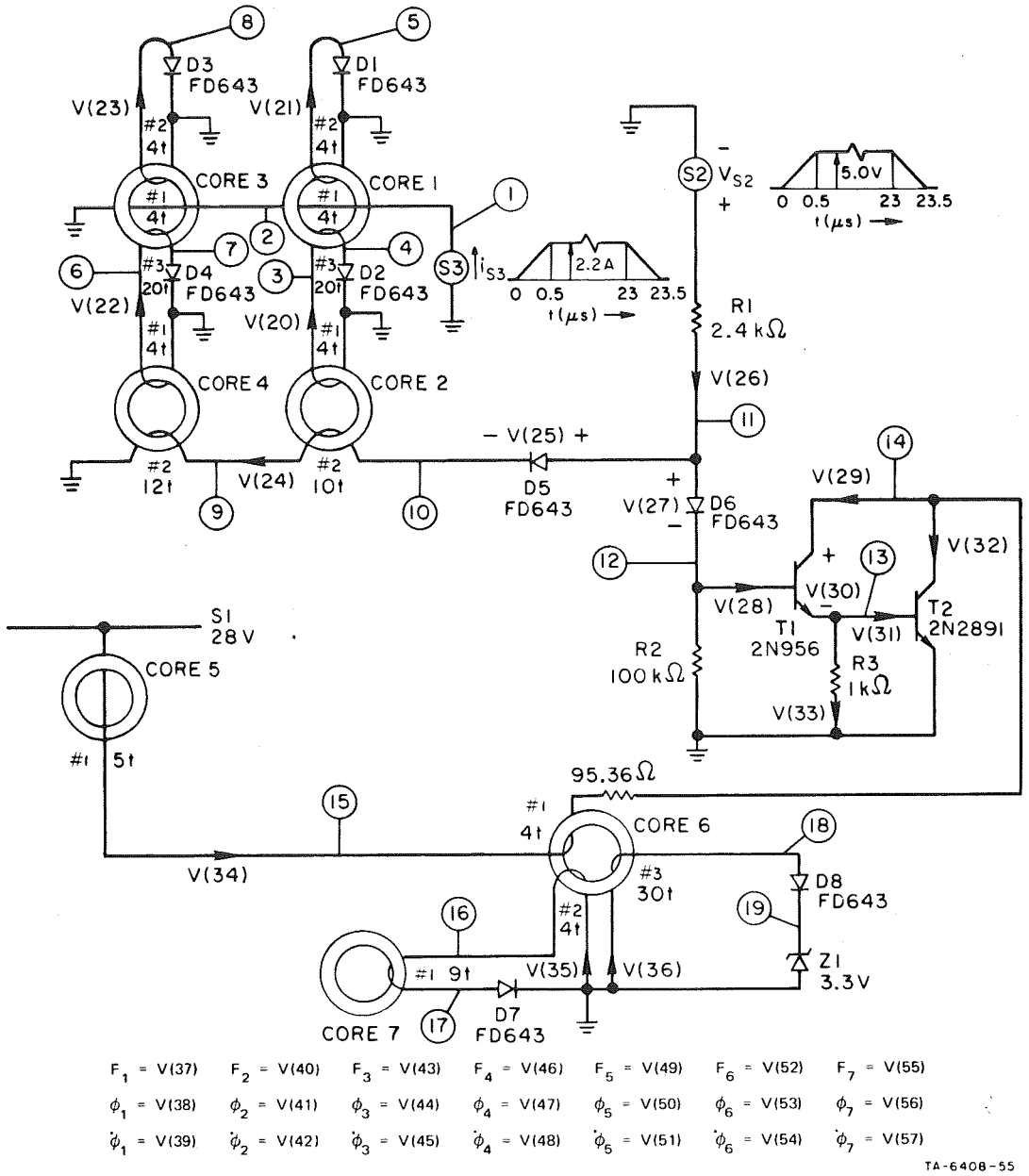


Figure 41 EXPERIMENTAL (heavy line) AND COMPUTED (light line) CURRENT AND $\dot{\phi}$ WAVEFORMS OF A TWO-PHASE CURRENT DRIVER AT T = 25° C IN MODE I [(a) - (d)] AND MODE II [(e) AND (f)]



F_1 = V(37) F_2 = V(40) F_3 = V(43) F_4 = V(46) F_5 = V(49) F_6 = V(52) F_7 = V(55)

ϕ_1 = V(38) ϕ_2 = V(41) ϕ_3 = V(44) ϕ_4 = V(47) ϕ_5 = V(50) ϕ_6 = V(53) ϕ_7 = V(56)

$\dot{\phi}_1$ = V(39) $\dot{\phi}_2$ = V(42) $\dot{\phi}_3$ = V(45) $\dot{\phi}_4$ = V(48) $\dot{\phi}_5$ = V(51) $\dot{\phi}_6$ = V(54) $\dot{\phi}_7$ = V(57)

TA-6408-55

Figure 42 AN INFORMATION-SENSING DRIVER CIRCUIT

generated voltage $V(10) = 10\dot{\phi}_2 - 12\dot{\phi}_4$ is high enough to block Diode $D5$. Consequently Diode $D6$ becomes unblocked and Transistors $T1$ and $T2$ turn on (become saturated). The resulting INFO current pulse, $V(34)$, sets the input core, Core 5, and clears the output core, Core 6. An output signal, $V(19)$, is thus generated.

If Cores 1 and 3 are initially in the same state (ONE or ZERO), then $\dot{\phi}_2 = \dot{\phi}_4$ and $V(10) < 0$. Consequently Diode D5 is unblocked and Diode $D6$ is either blocked or conducts a negligible current. Transistors $T1$ and $T2$ then remain off and Core 5 is not set (in addition, $\phi_6 = -\phi_r$).

The provisions to turn on Transistors $T1$ and $T2$ when initially $\phi_1 = -\phi_r$ and $\phi_3 = \phi_r$ are not shown in Fig. 42. A circuit identical with the one consisting of Resistor $R1$, Diodes $D5$ and $D6$, and the windings through Cores 2 and 4 is added in parallel, except that the numbers of turns are interchanged.

b. Input Data

In Fig. 42, we designate the sources, circuit elements, unknown nodal voltages, and auxiliary unknowns in a similar fashion as in Figs. 35 and 40. The circuit includes 3 sources (one of which is a current source), 1 zener diode, 8 diodes, 2 transistors, and 7 cores. There are 19 unknown voltages and 38 auxiliary unknowns, $V(20)$ through $V(57)$, as marked.

c. Results

Computed waveforms of $\dot{\phi}_1$, $\dot{\phi}_2$, $\dot{\phi}_4$, $\dot{\phi}_5$, $\dot{\phi}_6$, and $V_{out} = V(19)$ are shown in Fig. 43. No experimental oscillograms were available for comparison.

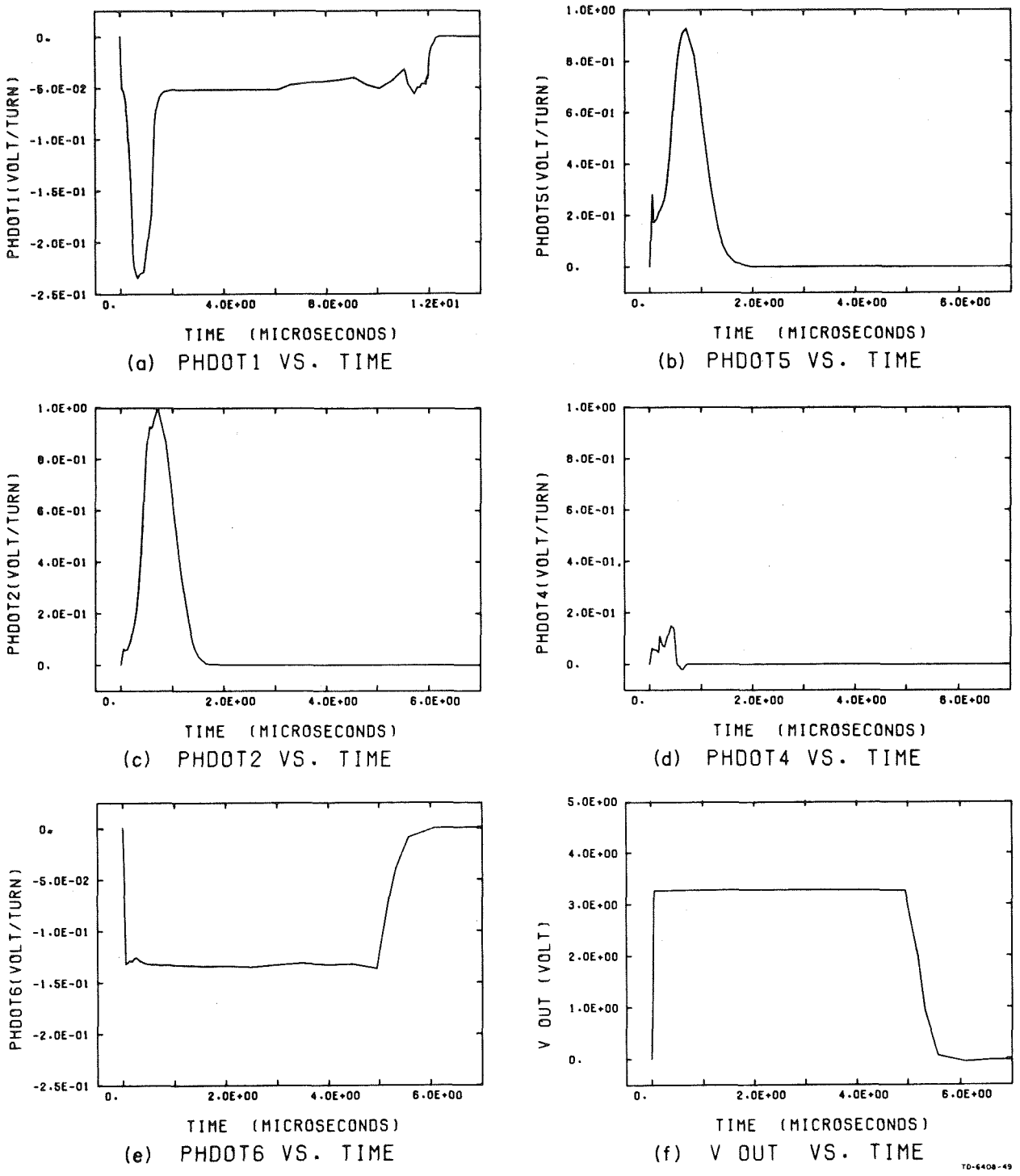


Figure 43 COMPUTED WAVEFORMS OF $\dot{\phi}_1$, $\dot{\phi}_2$, $\dot{\phi}_4$, $\dot{\phi}_5$, $\dot{\phi}_6$, AND V_{out} OF AN INFORMATION-SENSING DRIVER CIRCUIT FOR THE CASE OF 00000000001 STATE

REFERENCES

1. D. Nitzan, "Flux Switching in Multipath Cores," Report 1, for Jet Propulsion Laboratory, Contract 950095 under NASw-6, SRI Project 3696, Stanford Research Institute, Menlo Park, California (November 1961).

2. D. Nitzan and V. W. Hesterman, "Flux Switching in Multipath Cores," Report 2, for Jet Propulsion Laboratory, Contract 950095 under NASw-6, SRI Project 3696, Stanford Research Institute, Menlo Park, California (November 1962).

3. D. Nitzan and V. W. Hesterman, "Flux Switching in Multipath Cores," Report 3, for Jet Propulsion Laboratory, Contract 950095 under NASw-6, SRI Project 3696, Stanford Research Institute, Menlo Park, California (June 1964).

4. D. Nitzan and V. W. Hesterman, "Flux Switching in Multipath Cores," Report 4 for Jet Propulsion Laboratory, Contract 950943 under NAS7-100, SRI Project 5094, Stanford Research Institute, Menlo Park, California (July 1965).

5. D. Nitzan, V. W. Hesterman, and E. K. Van De Riet, "Flux Switching in Magnetic Circuits," Report 5, for Jet Propulsion Laboratory, Contract 951383 under NAS7-100, SRI Project 5670, Stanford Research Institute, Menlo Park, California (November 1966).

6. D. Nitzan, "Flux Switching in Overdriven Multipath Cores," *J. Appl. Phys.*, Pt. 2, Vol. 34, No. 4, pp. 1121-1122 (April 1963).

7. D. Nitzan, "Engineering Flux-Switching Models for Toroidal and Multipath Cores," *IEEE Trans. on Comm. & Elec.*, Vol. 83, pp. 309-314 (May 1964).

8. D. Nitzan, "Flux Division in a Saturable Multipath Core," *IEEE Trans. on Electronic Computers*, Vol. EC-13, No. 3, pp. 272-277 (June 1964).

9. V. W. Hesterman, "Switching Properties of a Partially-Set Square-Loop Ferrite Core," *IEEE Trans. on Magnetics*, Vol. MAG-1, No. 4, pp. 309-314 (December 1965).

10. D. Nitzan, "Computation of Flux Switching in Magnetic Circuits," *IEEE Trans. on Magnetics*, Vol. MAG-1, No. 3, pp. 222-234 (September 1965).

11. D. Nitzan, "Models for Elastic and Inelastic Flux Switching," *IEEE Trans. on Magnetics*, Vol. MAG-2, No. 4. pp. 751-760 (December 1966).

12. D. Nitzan and V. W. Hesterman, "Elastic Flux-Switching Properties of a Thin Ferrite Core," *J. Appl. Phys.*, Vol. 38, No. 3, pp. 1199-1200 (March 1967).

13. D. Nitzan, "A Square-Loop Magnetic Core Model for Computer-Aided Circuit Transient Analysis," *IEEE Trans. on Nuclear Science*, Vol. NS-15, No. 6 (December 1968).

14. D. Nitzan, "MTRAC—an Automated Computer Program for Analysis of Circuits Including Magnetic Cores," *1969 Proc. INTERMAG Conference*, p. 24.6.

15. W. Hochwald and C. T. Kleiner, "Digital Simulation of Nonlinear Electromagnetic Circuits," *IEEE Trans. on Magnetics* (1966 INTERMAG), Vol. MAG-2, No. 2, pp. 532-539 (September 1966).

16. C. T. Kleiner, E. D. Johnson, and W. D. Ashcraft, "A General Purpose System of Digital Codes for Linear/Nonlinear Network and System Analysis," *IEEE Intl. Convention Record*, March 20-23, 1967, Part 5, pp. 136-148.

17. C. T. Kleiner, G. Kinoshita, and E. D. Johnson, "Simulation and Verification of Transient Nuclear Radiation Effects in Semiconductor Electronics," *IEEE Trans. on Nuclear Science*, Vol. NS-11, No. 5 (November 1964).

18. E. D. Johnson, C. T. Kleiner, L. R. McMurray, E. L. Steele, and F. A. Vassallo, "Transient Radiation Analysis by Computer Program (TRAC)" Vol. I and Vol. II, Contract DAAG 39-68-C-0041, (Sponsored by U.S. Army Materiel Command, Harry Diamond Laboratories, and DASA), Autonetics, A Division of North American Rockwell, (June 1968).

19. Magnetics Inc., "Design Manual Featuring Tape Wound Cores," Catalog No. TWC-300, 1962.

20. J. Todd, *Survey of Numerical Analysis* (McGraw-Hill Book Company, Inc., New York, N.Y., 1962).

INDEX*

---

* Unlabeled page numbers correspond to Volume I. Page numbers preceded by the label "II." correspond to Volume II.

STANFORD
RESEARCH
INSTITUTE

Main Offices and Laboratories

333 Ravenswood Avenue
Menlo Park, California 94025
(415) 326-6200
Cable: STANRES, Menlo Park
TWX: 910-373-1246

Regional Offices and Laboratories

Southern California Laboratories
820 Mission Street
South Pasadena, California 91030
(213) 799-9501 • 682-3901
TWX: 910-588-3280

SRI-Washington
1611 North Kent Street, Rosslyn Plaza
Arlington, Virginia 22209
(703) 524-2053
Cable: STANRES, Washington, D.C.
TWX: 710-955-1137

SRI-New York
200 E. 42nd Street
New York, New York 10017
(212) 661-5313

SRI-Huntsville
Missile Defense Analysis Office
4810 Bradford Blvd., N.W.
Huntsville, Alabama 35805
(205) 837-3050
TWX: 810-726-2112

SRI-Chicago
10 South Riverside Plaza
Chicago, Illinois 60606
(312) 236-6750

SRI-Europe
Pelikanstrasse 37
8001 Zürich, Switzerland
Telephone: (051) 44 09 22
Cable: STANRES, Zürich

SRI-Scandinavia
Sveavagen 13-15
Box 1436
S-111 84 Stockholm, Sweden
Telephone: 23 03 10
Telex: 1617 STANRES S

SRI-Japan
Edobashi Building, 8th Floor
1-6, Nihonbashi Edobashi
Chuo-ku, Tokyo
Telephone: Tokyo 271-7108
Cable: STANRESEARCH, Tokyo

SRI-Southeast Asia
Bangkok Bank Building
182 Sukhumvit Road
Bangkok, Thailand
Telephone: 55292 or 58673
Cable: STANRES, Bangkok

Representatives

France
Roger Godino
94, Boulevard du Montparnasse
75 Paris 14e, France
Telephone: 633 37 30

Italy
Lorenzo L. Franceschini
Via Macedonio Melloni 49
20129 Milan, Italy
Telephone: 72 32 46

Philippines
Roberto V. Ongpin
SyCip, Gorres, Velayo & Co.
P.O. Box 589
Manila, Philippines
Telephone: 88 55 41
Cable: Certified

Portugal
J. Gasparinho Correia
Avenida João XXI, 22-3° Esq.
Lisbon, Portugal
Telephone: 72 64 87

January 1969